

A Mass Tracking Formulation for Bubbles in Incompressible Flow

Mridul Aanjaneya*, Saket Patkar*, Ronald Fedkiw*

Stanford University, 353 Serra Mall, Gates Computer Science Room 207, Stanford, CA 94305

Abstract

We devise a novel method for treating bubbles in incompressible flow that relies on the conservative advection of bubble mass and an associated equation of state in order to determine pressure boundary conditions inside each bubble. We show that executing this algorithm in a traditional manner leads to stability issues similar to those seen for partitioned methods for solid-fluid coupling. Therefore, we reformulate the problem monolithically. This is accomplished by first proposing a new fully monolithic approach to coupling incompressible flow to fully nonlinear compressible flow including the effects of shocks and rarefactions, and then subsequently making a number of simplifying assumptions on the air flow removing not only the nonlinearities but also the spatial variations of both the density and the pressure. The resulting algorithm is quite robust, has been shown to converge to known solutions for test problems, and has been shown to be quite effective on more realistic problems including those with multiple bubbles, merging and pinching, etc. Notably, this approach departs from a standard two-phase incompressible flow model where the air flow preserves its volume despite potentially large forces and pressure differentials in the surrounding incompressible fluid that should change its volume. Our bubbles readily change volume according to an isothermal equation of state.

1. Introduction

Numerical methods for simulating two-phase incompressible flows have received enormous attention popularized by [3, 33, 32]. Quite often, the air phase can be assumed to have small inertia relative to the water phase and need not be simulated, e.g., when studying the impact of ocean waves on a ship, which gives rise to free surface flows. However, such flows ignore effects like air entrainment and are unsuitable for modeling bubble dynamics. To prevent the air bubbles from collapsing unnaturally, methods have been proposed for computing a pressure inside these bubbles by tracking their volume over time and using an equation of state [30]. But volume can change radically if a bubble rises a long distance, merges with other nearby bubbles, or breaks up into smaller bubbles. Since the total mass of air inside water is conserved over time (ignoring mass exchange across the interface such as vaporization), we propose to track the bubble mass instead which avoids these difficulties.

We devised our method by first proposing a rather straightforward approach based on mass tracking in Section 2. This approach suffers from stability issues which have characteristics similar to partitioned (as opposed to monolithic) methods for solid-fluid coupling, see for example [13, 26, 12] and the references therein. These issues can be alleviated using outer iterations on the partitioned solver, as discussed in Section 2.4, although this can require ten or more Poisson solves per time step and is thus computationally expensive. Instead, we take a more traditional monolithic approach for the air-water problem similar to the solid-fluid coupling in [13, 26, 12] as motivated by [19]. We begin by revisiting the partitioned solver for incompressible and compressible flow from [5] and devise a monolithic solver using the ideas from [19] to couple together incompressible flow with fully nonlinear compressible flow including shocks and rarefactions. The results of this method are shown in Section 3 for both the gamma gas law and an isothermal equation

*{aanjaneya,patkar,fedkiw}@cs.stanford.edu, Stanford University

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 14 OCT 2012		2. REPORT TYPE		3. DATES COVERED 00-00-2012 to 00-00-2012	
4. TITLE AND SUBTITLE A Mass Tracking Formulation for Bubbles in Incompressible Flow				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Stanford University, Department of Computer Science, Stanford, CA, 94305				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT We devise a novel method for treating bubbles in incompressible flow that relies on the conservative advection of bubble mass and an associated equation of state in order to determine pressure boundary conditions inside each bubble. We show that executing this algorithm in a traditional manner leads to stability issues similar to those seen for partitioned methods for solid-fluid coupling. Therefore, we reformulate the problem monolithically. This is accomplished by first proposing a new fully monolithic approach to coupling incompressible flow to fully nonlinear compressible flow including the effects of shocks and rarefactions, and then subsequently making a number of simplifying assumptions on the air flow removing not only the nonlinearities but also the spatial variations of both the density and the pressure. The resulting algorithm is quite robust, has been shown to converge to known solutions for test problems, and has been shown to be quite effective on more realistic problems including those with multiple bubbles, merging and pinching, etc. Notably, this approach departs from a standard two-phase incompressible flow model where the air flow preserves its volume despite potentially large forces and pressure differentials in the surrounding incompressible fluid that should change its volume. Our bubbles readily change volume according to an isothermal equation of state.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 50	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

of state. Then in Section 4 we simplify this approach greatly so that it is in line with the straightforward bubble simulation method of Section 2. This is achieved by setting both the bubble density and the bubble pressure to be spatially constant, although time-varying.

In summary, our method for simulating bubbles in free surface incompressible flows is as follows. The incompressible flow is treated in a standard fashion with an unconditionally stable advection method [8, 27], an implicit method for viscosity, and the inclusion of surface tension using jump conditions as proposed in [22, 18]. For large air bodies connected to the atmosphere, we use a free surface condition of the atmospheric pressure as a Dirichlet boundary condition as is done in standard free surface incompressible flow solvers. For bubbles, we define mass inside each bubble and conservatively advect that mass using [21] with the air velocity, but also ensuring that the mass stays confined within the zero level set using ideas proposed in [20]. Density variations within the bubble are ignored setting the bubble density equal to the total bubble mass divided by the total bubble volume which is computed using a flood fill algorithm. The bubble pressure is then found using an equation of state $p = B\rho$, although simply applying this pressure as a Dirichlet boundary condition on the incompressible fluid leads to instabilities symptomatic of partitioned methods. Instead, we use the pressure evolution equation [10] and the ideas proposed in [19] in order to formulate a fully coupled solve between a single degree of freedom pressure for the bubble and the surrounding incompressible flow. This provides a very stable monolithic coupling for interactions between bubbles and the surrounding incompressible flow. The resulting pressure can be used to update the incompressible flow as usual, whereas the air velocities are updated by solving a second Poisson equation with Neumann boundary conditions forcing the bubble velocities to match the motion of the surrounding incompressible fluid including both expansion and contraction allowing the bubble to change volume if necessary. Section 6 gives a more detailed summary of our method referring to the appropriate equations throughout the text and highlights its efficacy with various examples - including the ability to treat multiple bubbles which may also split and merge.

2. A partitioned approach

We begin with a straightforward approach that conservatively advects the bubble mass and uses the isothermal equation of state $p = B\rho$ to compute a pressure inside the bubble which is subsequently used as a Dirichlet boundary condition for making the incompressible velocities divergence free. As we will see, this approach lacks stability properties because the current pressure in the bubble is not a good predictor of what the pressure would be in the next time step after applying the incompressible flow pressure to generate a new velocity field that changes the size of the bubble. If one couples the air pressure as a degree of freedom instead of as a Dirichlet boundary condition, then the incompressible Poisson solver can better react to the anticipated changes in the bubble volume. Thus, after presenting our straightforward approach in this section (which we call partitioned), we present a monolithic solver in Section 3.

2.1. Incompressible flow

The incompressible Navier-Stokes equations are given by

$$\vec{v}_t + (\vec{v} \cdot \nabla)\vec{v} + \frac{\nabla p}{\rho_I} = \frac{\nabla \cdot (\mu \nabla \vec{v})}{\rho_I} + \vec{f} \quad (1)$$

$$\nabla \cdot \vec{v} = 0 \quad (2)$$

where ρ_I is the density, \vec{v} is the velocity, μ is the coefficient of viscosity and \vec{f} is the net body force acting on the incompressible fluid. These equations are discretized on a MAC grid using the projection method [6], where we first explicitly update

$$\frac{\vec{v}^* - \vec{v}^n}{\Delta t} + (\vec{v} \cdot \nabla)\vec{v} = \frac{\nabla \cdot (\mu \nabla \vec{v})}{\rho_I} + \vec{f} \quad (3)$$

and then solve for the pressure via

$$\nabla \cdot \frac{\nabla p}{\rho_I} = \frac{\nabla \cdot \vec{v}^*}{\Delta t} \quad (4)$$

in order to update the intermediate velocity \vec{v}^* as follows

$$\frac{\vec{v}^{n+1} - \vec{v}^*}{\Delta t} + \frac{\nabla p}{\rho_I} = 0 \quad (5)$$

We use the level set method [24] to track the interface between the bubbles and the incompressible fluid. Before updating the incompressible velocities through advection, they are extrapolated across the interface in order to define ghost node values. This could be accomplished using constant extrapolation normal to the interface by solving the equation $I_\tau + \vec{N} \cdot \nabla I = 0$, in fictitious time τ for each component I of \vec{v} . We instead compute the steady state solution using the fast extension method of [1]. The incompressible velocities are then advected using semi-Lagrangian advection which can be made second order accurate using a MacCormack-style method as in [27]. The level set function ϕ is advected using the particle level set method of [7] and the semi-Lagrangian advection scheme of [8]. To keep the level set a signed distance function we use the modified fast marching method proposed in [23].

The treatment of viscosity for multiphase incompressible flow with appropriate jump conditions at the interface is discussed in [18]. However, viscosity is solved for explicitly in [18] which has a severe time step restriction of $\Delta t \propto O(\Delta x^2)$. In order to take large time steps, we consider an implicit treatment of viscosity. As discussed in [17], if all the jump conditions are treated implicitly then the equations for all components of the velocity are coupled together. Although one could take an approach similar to [25] for spatially varying viscosity where the coupling terms are treated explicitly and other terms are treated implicitly in order to get decoupling of various components, there can be some time step restrictions based on the jump conditions. For the simulation of bubbles, we assume that the dynamics inside the bubbles contain little momentum, hence, they cannot absorb any viscous momentum from the liquid. Thus, we enforce Neumann boundary conditions at the interface that the derivative of each component of the incompressible velocity is zero. Thus, the jump in pressure due to viscosity is also zero since the normal component of the viscous stress vanishes across the interface. Finally, as we assume constant viscosity in the incompressible fluid, the equations for the different components of the incompressible velocity decouple as well. In two spatial dimensions, equation (3) can be written component-wise as

$$\frac{v_1^* - v_1^n}{\Delta t} + (\vec{v} \cdot \nabla) v_1 = \frac{\nabla \cdot (\mu \nabla v_1^*)}{\rho_I} + f_1 \quad (6)$$

$$\frac{v_2^* - v_2^n}{\Delta t} + (\vec{v} \cdot \nabla) v_2 = \frac{\nabla \cdot (\mu \nabla v_2^*)}{\rho_I} + f_2 \quad (7)$$

The advection and external forces can be applied first to obtain \hat{v}_1 and \hat{v}_2 followed by a viscous solve of the form

$$\frac{v_1^* - \hat{v}_1}{\Delta t} = \frac{\nabla \cdot (\mu \nabla v_1^*)}{\rho_I} \quad (8)$$

$$\frac{v_2^* - \hat{v}_2}{\Delta t} = \frac{\nabla \cdot (\mu \nabla v_2^*)}{\rho_I} \quad (9)$$

Since \hat{v}_1 and \hat{v}_2 are not divergence free and the viscous update equations have been derived assuming the divergence free condition, \hat{v}_1 and \hat{v}_2 are sometimes first projected to be divergence free before applying viscosity. However, note that the pressure projection is not idempotent in the presence of a non-zero pressure gradient across the incompressible fluid. In this case we do not project \hat{v}_1 and \hat{v}_2 to be divergence free before the viscous update. Note that the advection terms are computed in a thin band of ghost cells so that there are adequate values when the interface moves, however, the viscous terms can only be updated interior to

the level set due to the need to prescribe interface boundary conditions. Therefore, the level set must be moved to its new time t^{n+1} location before applying the viscous update.

We use the second order cut-cell method of [11] for computing the pressure in equation (4) to make the incompressible velocities divergence free, where the pressure inside the bubbles and the outside air are used as Dirichlet boundary conditions. In the presence of surface tension, the term $\sigma\kappa$ is added to the Dirichlet pressure, where σ is the coefficient of surface tension and κ is the curvature of the interface, computed using the level set method [24]. See also Section 5.

The incompressible time step Δt_I is computed by enforcing the following inequality at every cell center, as described in [18],

$$\Delta t_I \left(\frac{C_{\text{cfl}} + \sqrt{C_{\text{cfl}}^2 + 4S_{\text{cfl}}^2 + 4F_{\text{cfl}}^2}}{2} \right) \leq 1 \quad (10)$$

Here, C_{cfl} accounts for the convection terms where v_1 and v_2 have been averaged to the cell center,

$$C_{\text{cfl}} = \frac{|v_1|}{\Delta x} + \frac{|v_2|}{\Delta y} \quad (11)$$

S_{cfl} accounts for the surface tension forces,

$$S_{\text{cfl}} = \sqrt{\frac{\sigma\kappa}{\rho_I(\min\{\Delta x, \Delta y\})^2}} \quad (12)$$

and F_{cfl} accounts for the body forces \vec{f}

$$F_{\text{cfl}} = \sqrt{\frac{|f_1|}{\Delta x} + \frac{|f_2|}{\Delta y}} \quad (13)$$

Note that the time step restrictions due to viscosity are not present as it is treated implicitly.

2.2. The oscillating bubble problem

We consider a model oscillating bubble problem in one spatial dimension as shown in Figure 1, where an air bubble of radius $r^0 = .1$ m with initial density $\rho^0 = 1.1$ kg/m³ is inside a water “sphere” of radius $r_w^0 = .4$ m. The computational domain is $[0 \text{ m}, 1 \text{ m}]$ which gives .1 m of free air on each side of the water region. Figure 2 shows the problem in two spatial dimensions where an air bubble of radius $r^0 = .1$ m with initial density $\rho^0 = 1.1$ kg/m³ is inside a water sphere of radius $r_w^0 = .4$ m. The computational domain is $[0 \text{ m}, 1 \text{ m}] \times [0 \text{ m}, 1 \text{ m}]$. The setup for the problem in three spatial dimensions is defined similarly. For simplicity, there is no gravity, surface tension or viscosity in the system. Since the bubble is slightly compressed with density $\rho^0 = 1.1$ kg/m³, there will be a larger pressure p^0 inside the bubble than in the ambient air which is taken to be a free surface condition of $p_{\text{atm}} = 101,325$ Pa and therefore, the bubble will start to expand, subsequently vibrating back and forth. The appendices derive a second order ODE given by equations (63), (70) and (77). In all three equations, we take the standard approach of solving for $\ddot{R}(t)$, rewriting the second order equation as a first order system, subsequently integrating in time using third order accurate TVD Runge-Kutta, and refining the time step until the solutions converge to obtain data that we use for the “exact” solutions when these equations are considered.

For this problem, we modify the time step restriction of Section 2.1 to account for velocities near zero when the bubble volume is at an extrema, i.e., when it has maximum or minimum volume. To prevent the time step from becoming excessively large in these cases, we add a term to Δt_I that estimates the change in velocity over a time step, similar in spirit to what was done in [18] for body forces and [19] for compressible flow - see also Section 3.4 for a quick summary. Essentially what is needed is an estimate for $|\nabla p|$ which will influence the velocity. We do this by computing l_x and l_y as the minimum thicknesses of the water region

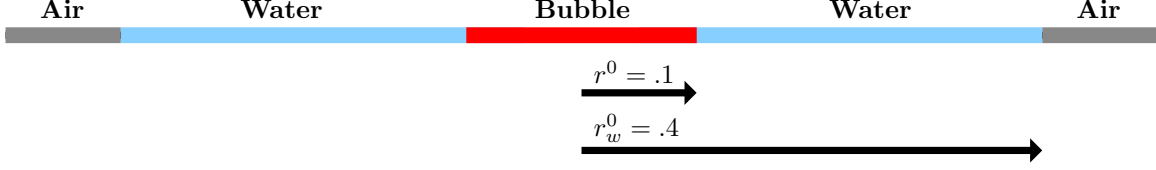


Figure 1: Setup for the oscillating bubble problem in one spatial dimension.

in the x and y directions, and approximate $|p_x|$ as $|p_{\text{atm}} - p^n|/l_x$ and $|p_y|$ as $|p_{\text{atm}} - p^n|/l_y$, where p^n is the pressure inside the bubble at time t^n . Then we write

$$F_{\text{cfl}} = \sqrt{\frac{|p_{\text{atm}} - p^n|}{\rho^n l_x \Delta x} + \frac{|p_{\text{atm}} - p^n|}{\rho^n l_y \Delta y}} \quad (14)$$

and include this in equation (10) analogous to equation (53).

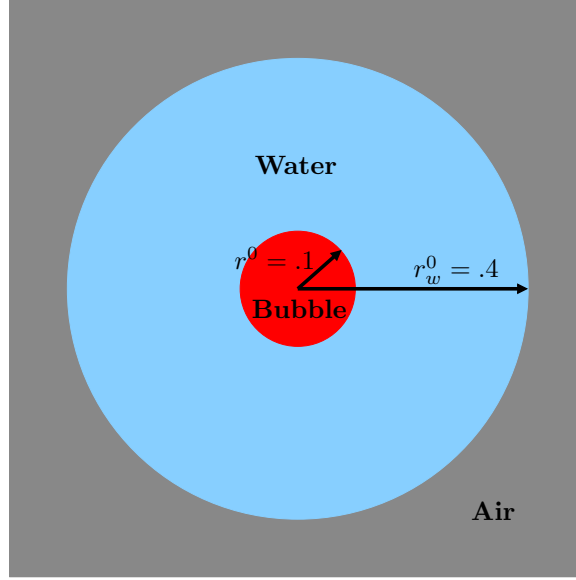


Figure 2: Setup for the oscillating bubble problem in two spatial dimensions.

2.3. Treating bubbles

Initially, each bubble is assigned a density (or mass), and the density is advected using the unconditionally stable conservative semi-Lagrangian scheme of [21]. This scheme is especially effective in keeping track of small bubbles, since the level set loses volume over time and cannot keep track of sub-grid level details. Although one could advect the bubble density using velocities extrapolated from the incompressible flow, we use a more accurate approach where air velocities are constructed and maintained in a separate velocity field \vec{u} . The velocity used for level set advection is a hybrid between the incompressible flow velocity and the air velocities. Even so, numerical smearing and other errors will cause the location of the zero level set and the location of the non-zero bubble densities to drift apart over time. We address this as follows. First we compute the total mass that belongs to a bubble as the sum of all the mass inside the bubble and all the mass closest to that bubble. Then we use a flood fill algorithm on that bubble to identify all grid cells belonging to that connected component. The volume of this connected component is carefully computed using a

piecewise linear reconstruction of the level set as outlined in [20]. The mass is then uniformly redistributed inside the bubble to obtain a spatially constant density ρ_b . The Dirichlet boundary condition used for the incompressible Poisson solve given in equation (4) is computed using the equation of state $p = B\rho$ which simplifies to $p = BM/V$, where M and V are the total mass and total volume of a bubble respectively. Here, B is taken to be $p_{\text{atm}} \text{ m}^3/\text{kg}$, so that a density of 1 kg/m^3 yields a pressure of $p_{\text{atm}} = 101,325 \text{ Pa}$.

Air velocities are treated in a manner similar to the incompressible flow velocities. Ghost node values are defined using the fast extension method of [1] exactly as is done for the incompressible velocities, and then the air velocities are advected using the MacCormack method [27]. Since the air is not strictly volume preserving as bubbles can expand and contract, we do not make the air flow incompressible as would be done for a truly two-phase incompressible flow. Instead, we solve a modified form of equations (4) and (5) for each bubble

$$\nabla \cdot \nabla \tilde{p} = \nabla \cdot \vec{u}^* - \nabla \cdot \vec{u}^{n+1} \quad (15)$$

$$\vec{u}^{n+1} - \vec{u}^* + \nabla \tilde{p} = 0 \quad (16)$$

where $\tilde{p} = \Delta t p / \rho_b$, ρ_b is the spatially constant air density inside the bubble (which could be different per bubble), and \vec{u}^* is the post-advected air velocity. Equations (4) and (5) are solved first for the full water volume using Dirichlet boundary conditions, after which the resulting water velocities surrounding the bubble are used as Neumann boundary conditions to solve equation (15). Since the integral of the incompressible velocity around the surface of a bubble may not be zero, we compute the net divergence for the boundary of the bubble summing all these velocities divided by the number of cells in the bubble, and use that value for $\nabla \cdot \vec{u}^{n+1}$. This allows bubbles to expand and contract and otherwise change volume as they follow and are enslaved by the surrounding incompressible flow which has much higher momentum. Also note that the Poisson matrix resulting from equation (15) has a rank-deficiency of 1 due to the full Neumann boundary conditions and although the addition of $\nabla \cdot \vec{u}^{n+1}$ guarantees that the right hand side is in the range of the Poisson matrix, one still needs to take care to compute the minimum norm solution during the conjugate gradient solve.

In Figures 3 and 4 we present numerical profiles generated by the partitioned scheme. Consider the setup shown in Figure 1 and assume the bubble is initially at rest, i.e., the initial air density inside it is $\rho^0 = 1 \text{ kg/m}^3$. Analytically, the bubble should just stay at rest and the bubble volume should remain constant over time. We refer to this problem as the *stationary bubble problem*. Figure 3 shows the bubble volume profiles over time with the proposed partitioned scheme. Note that although the initial solution computed by this scheme is indeed constant, it quickly goes unstable at lower grid resolutions because of the forward Euler characteristic of the partitioned scheme. For higher grid resolutions, the nonphysical growth is slower as expected. For the one dimensional oscillating bubble problem, Figure 4 shows the resulting volume profiles over time under grid refinement. Again, note that the partitioned scheme shows instability at lower grid resolutions while converging under grid refinement. The results for the two dimensional simulations are similar.

2.4. An iterative approach

Although the partitioned scheme proposed in Section 2 converges to correct analytical solutions under grid refinement, it shows instability at lower grid resolutions because of its forward Euler characteristic. The most logical next step is to try using TVD Runge-Kutta methods for greater stability [28]. We approach this by applying second order accurate TVD Runge-Kutta on the bubble pressure p_b which is used as a Dirichlet boundary condition in equation (4). Specifically, we take two full steps of our method to compute the spatially constant density $\hat{\rho}_b^{n+2}$ inside the bubble and use the average $(\rho_b^n + \hat{\rho}_b^{n+2})/2$ for computing the Dirichlet pressure $p_b = B(\rho_b^n + \hat{\rho}_b^{n+2})/2$. Subsequently, we rewind the simulation to the beginning of the time step and use this Dirichlet pressure boundary condition in the incompressible flow solve in equations (4) and (5) in order to obtain the divergence free incompressible velocity field. Note that intermediate substeps can result in velocities that dictate a smaller step size than that chosen at the beginning of the time step. Ignoring this can result in inaccurate solutions, so if this occurs we revert to the beginning of the time step and start over using smaller step size. Figure 5 shows a comparison between the forward Euler scheme,

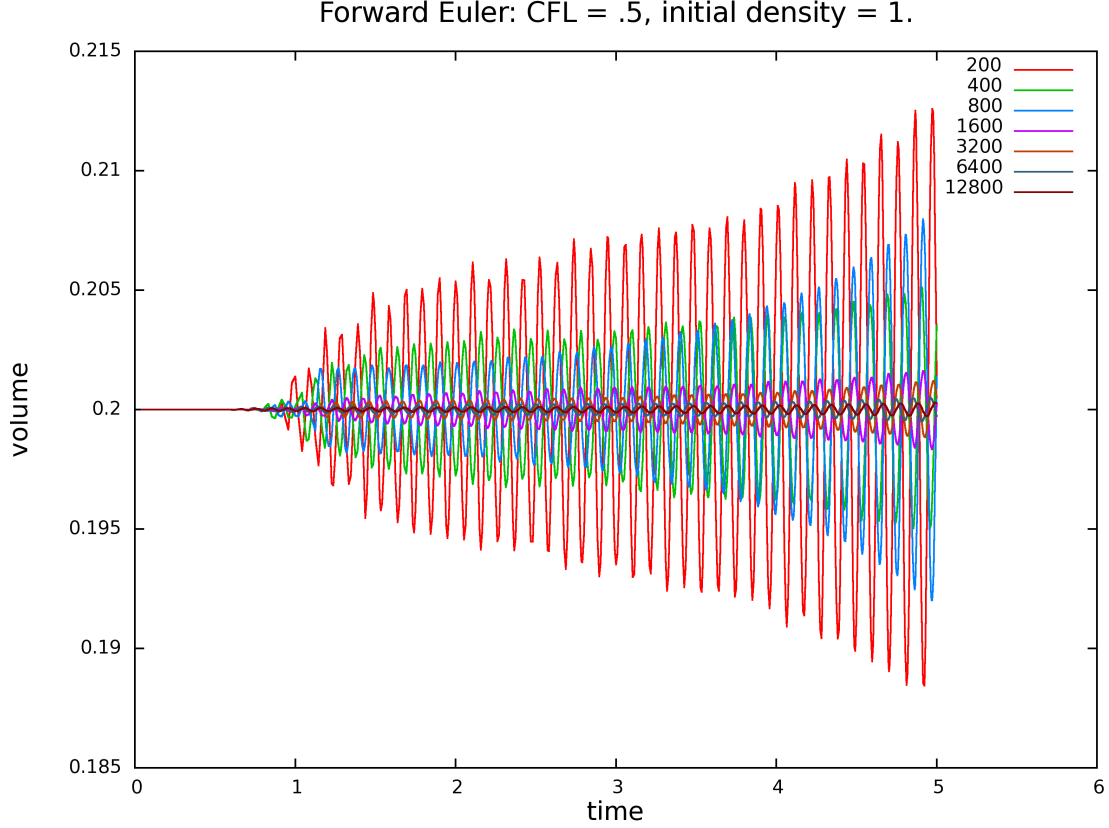


Figure 3: Numerical profiles generated by the partitioned scheme in one spatial dimension for bubble volume over time for the stationary bubble problem. Note the forward Euler characteristic of the partitioned scheme which causes instability in the computed solutions at lower grid resolutions. For higher grid resolutions, the nonphysical growth is slower as expected.

second order accurate TVD Runge-Kutta on the Dirichlet pressure p_b without modifying the time step, and the modified second order accurate TVD Runge-Kutta which reverts the simulation to the beginning of the time step if the CFL condition is violated. Note the intermittent spikes generated by the standard second order accurate TVD Runge-Kutta scheme, which are not present in the modified rewinding version as it obeys the CFL time step restriction even at intermediate steps. Also note that the modified second order accurate TVD Runge-Kutta scheme does a much better job at tracking the constant solution than the forward Euler version of the scheme.

The enhanced stability shown by the use of a second order accurate TVD Runge-Kutta scheme on the Dirichlet pressure boundary condition p_b used for the bubble in solving equations (4) and (5) motivates consideration of a technique similar to [19] where our goal is to replace the $p = BM/V$ Dirichlet boundary condition with the pressure the bubble would have in the next time step after solving for the incompressible velocities and advecting the bubble forward in time. Consider the pressure evolution equation [10],

$$p_t + \vec{u} \cdot \nabla p = -\rho c^2 \nabla \cdot \vec{u} \quad (17)$$

Since we assume that the air density inside the bubble is spatially constant, it follows from $p = B\rho$ that $\nabla p = 0$ and thus

$$p_t = -\rho c^2 \nabla \cdot \vec{u} \quad (18)$$

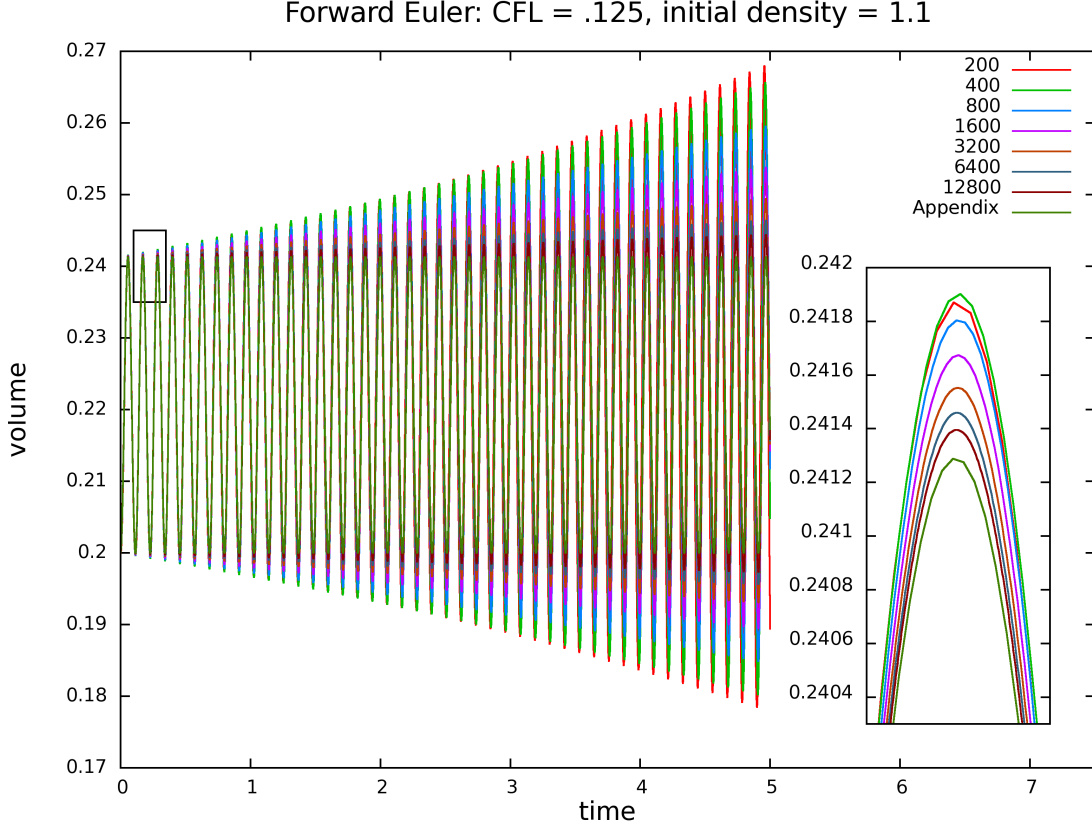


Figure 4: Numerical profiles generated by the partitioned scheme in one spatial dimension for bubble volume over time for the oscillating bubble problem. Note that the partitioned scheme shows instability at lower at lower grid resolutions because of its forward Euler characteristic, although converges to the “exact” solution under grid refinement.

For a gas governed by the equation of state $p = B\rho$, the sound speed c is defined as

$$c = \sqrt{p_\rho + \frac{pp_e}{\rho^2}} = \sqrt{B} \quad (19)$$

implying that it is constant in time and space. Discretizing equation (18) in time gives

$$p^{n+1} = p^n - \Delta t \rho^{n+1} B \nabla \cdot \vec{u} \quad (20)$$

where we set ρ to time t^{n+1} . As mass is conserved over time $M^{n+1} = M^n = M$. Using $p = B\rho$, $\rho^n = M/V^n$ and $\rho^{n+1} = M/V^{n+1}$, we obtain

$$\frac{BM}{V^{n+1}} - \frac{BM}{V^n} = -\frac{BM}{V^{n+1}} \Delta t \nabla \cdot \vec{u} \quad (21)$$

or

$$\Delta V = V^n \Delta t \nabla \cdot \vec{u} \quad (22)$$

where $\Delta V = V^{n+1} - V^n$. Using the divergence theorem, $V^n \nabla \cdot \vec{u}$ inside the bubble is equivalent to $\oint \vec{u} \cdot \vec{n} dS$. Letting \bar{u} be the average normal velocity on the boundary of the bubble and \mathcal{P} be the perimeter of the bubble allows us to write

$$\Delta V = \Delta t \mathcal{P} \bar{u} \quad (23)$$

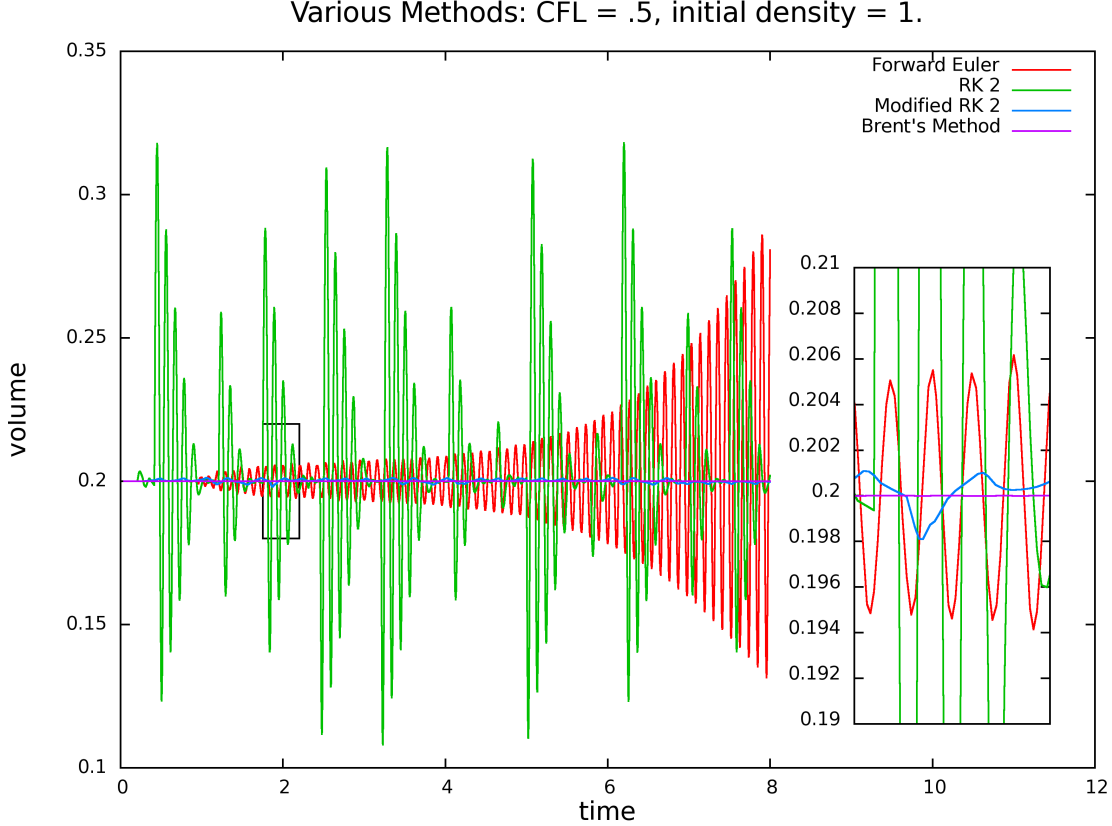


Figure 5: A comparison between numerical profiles for the one dimensional stationary bubble problem generated using the forward Euler scheme (red), second order TVD Runge-Kutta (green), modified second order TVD Runge-Kutta which takes time step restrictions imposed by intermediate velocities into account (blue), and Brent's method (magenta).

Note that we use a piecewise linear reconstruction of the level set as described in [20] for computing \mathcal{P} and \bar{u} .

The initial guess for the iterative solver sets $p = BM/V$ as a Dirichlet boundary condition for projecting the incompressible velocities. These velocities are then used to guess the bubble's volume V^{n+1} using equation (23). Since mass is constant, this predicts a new $p^{n+1} = B\rho^{n+1} = BM/V^{n+1}$. This pressure can again be set as a Dirichlet boundary condition to project the incompressible velocities and improve the guess for p^{n+1} . Through the iterative solver, we are looking for a fixed point for this “function”, i.e., $p = f(p)$ or a root of $g(p) = f(p) - p$. Note that if the input pressure is too large, the bubble expands and the predicted pressure drops, and similarly if the input pressure is too small, the bubble contracts and the predicted pressure increases. This allows us to place bounds on the solution. Basically, we start with $p^n = BM/V^n$ and if $f(p)$ is bigger, the initial guess for p is the left end-point of our interval. Otherwise, if $f(p)$ is smaller, this is taken as the right end-point of the interval. As the iteration proceeds, we eventually identify both left and right end-points. Once we find a bounding interval we use Brent's method [2] to find the root.

Figures 6(a) and 6(b) show the numerical profiles generated by the iterative solver for the bubble volume over time with increasing grid resolutions in one and two spatial dimensions for the stationary bubble problem. Note the improved stability achieved by the iterative solver. For the sake of comparison, the iterative solver is also shown labeled as Brent's method in Figure 5. Next consider the oscillating bubble problem. Figures 6(c) and 6(d) show the corresponding profiles for the bubble volume over time under grid refinement. Comparing Figures 4 and 6, note that the explicit method diverges for low grid resolutions

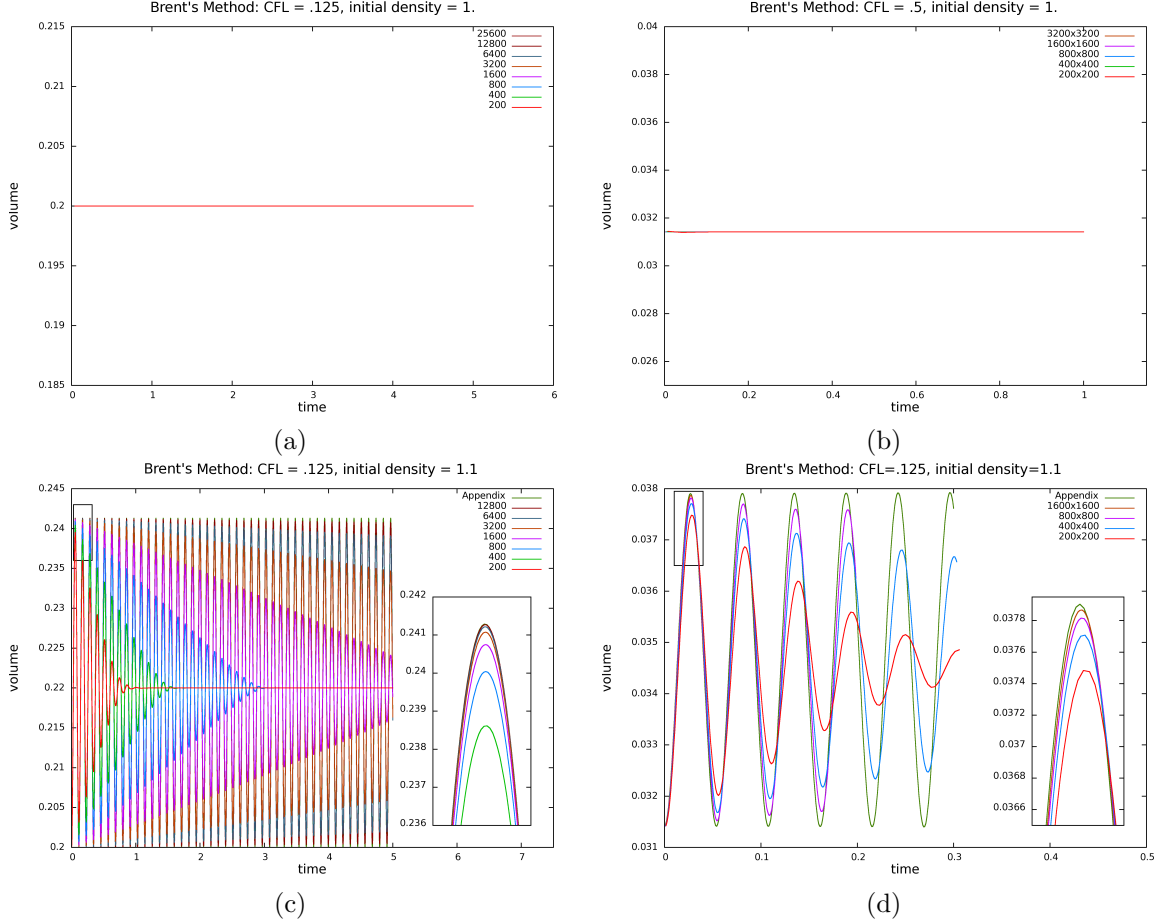


Figure 6: Numerical profiles generated by the iterative scheme for bubble volume over time under grid refinement for, (a) the stationary bubble problem in 1D, (b) the stationary bubble problem in 2D, (c) the oscillating bubble problem in 1D, (d) the oscillating bubble problem in 2D. Note the improved stability achieved by the iterative method as compared to the forward Euler scheme.

and eventually converges to the “exact” solutions as the grid is refined. In contrast, the implicit solution overly damps the phenomena on coarse grids while still converging to the “exact” solutions as the grid is refined. Obviously one would prefer the implicit approach over the explicit one so that although unresolved phenomena are not accurately resolved, i.e., the solution is overdamped, they at least do not explode and corrupt the entire solution.

3. Coupling compressible and incompressible flow

Motivated by the method in [19] which proposes an elliptic solver for pressure for compressible flow, our goal is to develop a fully monolithic solver which solves for air and water pressures together. However, before doing this we first develop a full monolithic solver that couples incompressible flow with fully non-linear compressible flow including contact discontinuities like shocks and rarefactions building upon the partitioned solver of [5]. They used a fully explicit scheme for the compressible fluid, and the pressure in the compressible region was used as a Dirichlet boundary condition when solving the incompressible Poisson equation for updating the incompressible velocities. Near the interface, the ghost fluid method (GFM) [9] was used to treat the boundary conditions in a manner that admitted sharp discontinuities while still allowing

for smooth discretizations across the interface. To achieve this, the interface values of pressure and normal velocity were carefully determined noting that, although these variables are continuous, they may possess kinks across the interface. Small errors in the normal velocity of the incompressible fluid create small errors in its divergence, which in turn can lead to large spurious pressure oscillations in the incompressible region. While small errors in the velocity of the compressible fluid cause the same small errors in its density, these have little effect on the gas since the gamma gas law equation of state is rather robust. Again, since the incompressible flow's pressure response is rather stiff, one can expect large variations in the incompressible pressure near the interface which in turn can lead to poor predictions of the interface pressure. While these errors in the interface pressure have a relatively small effect on the heavier incompressible fluid, they can have a rather large effect on the lighter compressible gas. Conversely, since the gamma gas law equation of state is rather robust, the compressible pressure tends to be smooth near the interface and is therefore a good candidate for the interface pressure. In view of these statements, [5] proposed using the incompressible region to determine the interface normal velocity and the compressible region to determine the interface pressure. In the presence of surface tension, the compressible pressure is not used directly, but is first modified according to the appropriate $[p] = \sigma\kappa$ jump condition. Although the method of [5] works well, it suffers from a strict time step restriction because the sound speed in the compressible fluid dictates the size of the time step. In addition, the method of [5] utilizes a partitioned coupling approach which can suffer from stability issues.

3.1. A semi-implicit formulation for compressible flow

Let ρ be the density of the compressible fluid, \vec{u} its velocity, E the total energy, p the pressure, and $\mathbf{U} = (\rho, \rho\vec{u}, E)$ the compressible state vector. The inviscid compressible Euler equations in multiple spatial dimensions are as follows

$$\mathbf{U}_t + \nabla \cdot \mathbf{F}(\mathbf{U}) = \begin{pmatrix} \rho \\ \rho\vec{u} \\ E \end{pmatrix}_t + \nabla \cdot \begin{pmatrix} \rho\vec{u} \\ \rho\vec{u} \otimes \vec{u} + p \\ (E + p)\vec{u} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (24)$$

A semi-implicit formulation for solving these equations was recently proposed in [19], where the flux vector $\mathbf{F}(\mathbf{U})$ was split into an advection part and a nonadvection part

$$\mathbf{F}_1(\mathbf{U}) = \begin{pmatrix} \rho\vec{u} \\ \rho\vec{u} \otimes \vec{u} \\ E\vec{u} \end{pmatrix}, \mathbf{F}_2(\mathbf{U}) = \begin{pmatrix} 0 \\ p \\ p\vec{u} \end{pmatrix} \quad (25)$$

The advection part $\mathbf{F}_1(\mathbf{U})$ was integrated explicitly to get intermediate values ρ^* , $(\rho\vec{u})^*$ and E^* , and since pressure does not affect the continuity equation, it follows that $\rho^{n+1} = \rho^*$. The original method of [19] used a modified ENO scheme for the explicit update to avoid Gibbs phenomena. Recently, [12] proposed using the standard ENO scheme [29] and a different method for computing the post-advected pressure (p_a , see below) which avoids this issue. We follow this improved approach. The nonadvection momentum and energy updates are

$$\frac{(\rho\vec{u})^{n+1} - (\rho\vec{u})^*}{\Delta t} = -\nabla p \quad (26)$$

$$\frac{E^{n+1} - E^*}{\Delta t} = -\nabla \cdot (p\vec{u}) \quad (27)$$

Motivated by the standard incompressible flow formulation, equation (26) is divided by ρ^{n+1} to obtain

$$\vec{u}^{n+1} = \vec{u}^* - \Delta t \frac{\nabla p^{n+1}}{\rho^{n+1}} \quad (28)$$

and its divergence is taken to obtain

$$\nabla \cdot \vec{u}^{n+1} = \nabla \cdot \vec{u}^* - \Delta t \nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) \quad (29)$$

Then the pressure evolution equation (17) is semi-discretized by fixing $\nabla \cdot \vec{u}$ to time t^{n+1} through the time step and by treating the advection terms explicitly. Let $e = E/\rho - \vec{u} \cdot \vec{u}/2$ denote the internal energy per unit mass, then the advected pressure is computed as $p_a = p^* = p(\rho^*, e^*)$ using the equation of state. Substituting p^* into the semi-discretized form of equation (17), we obtain

$$p^{n+1} = p^* - \Delta t \rho c^2 \nabla \cdot \vec{u}^{n+1} \quad (30)$$

Eliminating $\nabla \cdot \vec{u}^{n+1}$ by combining equations (30) and (29) and rearranging the terms gives

$$p^{n+1} - \Delta t^2 \rho^n (c^2)^n \nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = p^* - \Delta t \rho^n (c^2)^n \nabla \cdot \vec{u}^* \quad (31)$$

where the term ρc^2 has been fixed to the time t^n value. By composing the $\rho^n (c^2)^n$ terms into a diagonal matrix $P = [\Delta t^2 \rho^n (c^2)^n]$ and discretizing the gradient and divergence operators, we obtain the following system of equations

$$[P^{-1} + G^T (\hat{\rho}^{n+1})^{-1} G] \hat{p}^{n+1} = P^{-1} \hat{p}^* + G^T \hat{u}^* \quad (32)$$

where G is the discretized gradient operator, $-G^T$ the corresponding discretized divergence operator. The pressure is scaled by Δt , i.e., $\hat{p} = p \Delta t$, $\hat{\rho}$ is the density interpolated to cell faces and \hat{u} denotes a density-weighted averaged face velocity, i.e.,

$$\hat{\rho}_{i+1/2}^{n+1} = \frac{\rho_i^{n+1} + \rho_{i+1}^{n+1}}{2}, \quad \hat{u}_{i+1/2}^* = \frac{(\rho u)_i^* + (\rho u)_{i+1}^*}{\rho_i^{n+1} + \rho_{i+1}^{n+1}} \quad (33)$$

Note that the resulting matrix in equation (32) has an identity term in it, which allows fast solvers like preconditioned conjugate gradient (PCG) to converge in relatively few iterations. After solving equation (32) to obtain cell-centered pressure values, they are applied in a conservative flux-based manner to update the intermediate momentum and energy. Face pressures are computed using density-weighted averaging, i.e.,

$$\hat{p}_{i+1/2}^{n+1} = \frac{\rho_i^{n+1} \hat{p}_{i+1}^{n+1} + \rho_{i+1}^{n+1} \hat{p}_i^{n+1}}{\rho_i^{n+1} + \rho_{i+1}^{n+1}} \quad (34)$$

and face velocities are computed by rewriting equation (28) using face-averaged quantities as defined above.

$$\hat{u}_{i+1/2}^{n+1} = \hat{u}_{i+1/2}^* - (\hat{\rho}_{i+1/2}^{n+1})^{-1} G_{i+1/2} \hat{p}^{n+1} \quad (35)$$

Here, $G_{i+1/2}$ denotes the row of G corresponding to face $i + 1/2$. The flux-based implicit update then takes the form

$$(\rho \vec{u})_i^{n+1} = (\rho \vec{u})_i^* - \frac{\hat{p}_{i+1/2}^{n+1} - \hat{p}_{i-1/2}^{n+1}}{\Delta x}, \quad E_i^{n+1} = E_i^* - \frac{\hat{p}_{i+1/2}^{n+1} \hat{u}_{i+1/2}^{n+1} - \hat{p}_{i-1/2}^{n+1} \hat{u}_{i-1/2}^{n+1}}{\Delta x} \quad (36)$$

3.2. Explicit coupling step

We use the level set method to track the interface between the compressible and incompressible fluids. For the explicit part of the method, the interface boundary conditions are treated as described in [5]. Various quantities need to be extrapolated across the interface in either direction to define ghost node values. This is accomplished using constant extrapolation normal to the interface by solving the equation $I_\tau + \vec{N} \cdot \nabla I = 0$, in fictitious time τ for the different quantities I - we use the fast extension method of [1]. To compute the compressible ghost node state we decompose the extrapolated state vector into entropy, pressure and velocity, compute the cell centered incompressible velocity, and replace the normal component of the compressible velocity field with the normal component of the incompressible cell centered velocity field before reassembling entropy, pressure and velocity to obtain a ghost state for the compressible flow. The explicit update for the

compressible fluid consists of applying only the advection fluxes from equation (25) using these ghost node values. In order to properly handle uncovered cells a band of ghost node values is also updated in time.

The incompressible flow update proceeds similarly as to what was described in Section 2.1. The full incompressible velocity field is extrapolated across the interface into the compressible region to obtain ghost node values, the advection terms are updated to obtain intermediate velocities, and the viscous terms are updated noting that the compressible fluid is inviscid and thus cannot absorb any viscous stress.

Note that when applying viscosity to the incompressible fluid, the time \hat{t} velocities \hat{v}_1 and \hat{v}_2 are sometimes first projected to be divergence free, as noted in Section 2.1 after equations (8) and (9). In our monolithic approach, the divergence free projection is two-way coupled between the compressible and incompressible flow as given in equations (43) and (44) below in Section 3.3. Thus, our strategy is to apply this two-way coupled projection first and then use the resulting incompressible state to add the effects of the viscous terms to the incompressible fluid. Interestingly, note that this first coupled projection solve essentially gives the answer one would obtain if the incompressible flow was inviscid. Thus for the viscous case, we essentially obtain the inviscid solution first, rewind the compressible state vector \mathbf{U} to its pre-projected state, apply the viscous update to the incompressible region, and then apply the coupled projection once again to obtain a final solution which includes the effects of viscosity on the incompressible fluid.

3.3. Implicit coupling step

For the sake of exposition, we describe the implicit step in one spatial dimension. Multiple spatial dimensions are handled in a straightforward manner using a dimension-by-dimension approach. Consider the situation depicted in Figure 7. Let p_{int} denote the interface pressure and $\theta = |\phi(x_2)|/(|\phi(x_2)| + |\phi(x_3)|)$ be the cell fraction between the interface and the center of cell 2. Discretizing the incompressible flow Poisson equation (4) for cell 3, we obtain

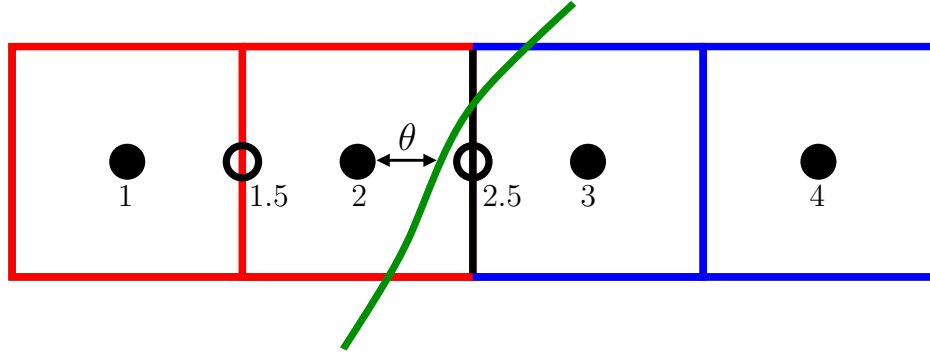


Figure 7: A global Poisson solve for coupling together compressible and incompressible fluids. Compressible cells are shown with red borders, incompressible cells are shown with blue borders. The interface is shown in green and the shared face is colored black.

$$-\frac{\Delta t}{\Delta x} \left(\frac{p_4^{n+1} - p_3^{n+1}}{\rho_I \Delta x} - \frac{p_3^{n+1} - p_{\text{int}}}{\rho_I (1 - \theta) \Delta x} \right) = -\nabla \cdot \vec{v}^* \quad (37)$$

Let I and C subscripts represent values from the incompressible and compressible sides of the interface. For inviscid flow $\nabla p / \rho$ is continuous across the interface [18] satisfying

$$\frac{\nabla p_I}{\rho_I} = \frac{\nabla p_C}{\rho_C} \quad (38)$$

Although [18] showed this for two-phase incompressible flow, one can still derive equation (1) for compressible flow from the equations for conservation of mass and momentum and so it also holds for inviscid compressible

flow and mixed compressible-incompressible flow (as long as the strong form of the equations hold and derivatives exist). For viscous flows things can be more complex, however, in our case we assume the compressible flow is inviscid and that there is no viscous momentum transfer from the incompressible to the compressible flow as mentioned in Section 2.1 (see also Section 3.2) - allowing equation (38) to still hold. Approximating equation (38) with one-sided differences, we obtain

$$\frac{p_{\text{int}} - p_2^{n+1}}{\rho_C \theta \Delta x} = \frac{p_3^{n+1} - p_{\text{int}}}{\rho_I (1 - \theta) \Delta x} \quad (39)$$

where we take $\rho_C = \rho_2^{n+1}$ (one might also conceivably use $\hat{\rho}_{2.5}^{n+1}$). Solving equation (39) for the interface pressure p_{int} gives

$$p_{\text{int}} = \frac{\theta \rho_C p_3^{n+1} + (1 - \theta) \rho_I p_2^{n+1}}{\theta \rho_C + (1 - \theta) \rho_I} \quad (40)$$

Writing

$$\tilde{\rho} = \theta \rho_C + (1 - \theta) \rho_I \quad (41)$$

allows us to write

$$\frac{p_{\text{int}} - p_2^{n+1}}{\rho_C \theta \Delta x} = \frac{p_3^{n+1} - p_2^{n+1}}{\tilde{\rho} \Delta x} = \frac{p_3^{n+1} - p_{\text{int}}}{(1 - \theta) \rho_I \Delta x} \quad (42)$$

and thus, equation (37) becomes

$$-\frac{\Delta t}{\Delta x} \left(\frac{p_4^{n+1} - p_3^{n+1}}{\rho_I \Delta x} - \frac{p_3^{n+1} - p_2^{n+1}}{\tilde{\rho} \Delta x} \right) = -\nabla \cdot \vec{v}^* \quad (43)$$

Next consider the compressible Poisson equation in cell 2. Note that cell 3 has a valid compressible state vector after advection because we also update a band of ghost cells near the interface (see Section 3.2). Let $\hat{\rho}_{2.5}^{n+1}, \hat{\rho}_{1.5}^{n+1}$ denote the interpolated face density obtained by averaging cell-centered values. Discretizing the compressible Poisson equation (32) for cell 2 and using equation (42) gives

$$\frac{p_2^{n+1}}{\rho_2^n (c_2^n)^n \Delta t} - \frac{\Delta t}{\Delta x} \left(\frac{p_3^{n+1} - p_2^{n+1}}{\tilde{\rho} \Delta x} - \frac{p_2^{n+1} - p_1^{n+1}}{\hat{\rho}_{1.5}^{n+1} \Delta x} \right) = \frac{p_2^*}{\rho_2^n (c_2^n)^n \Delta t} - \nabla \cdot \hat{u}^* \quad (44)$$

Note that equations (43) and (44) together form a symmetric positive definite (SPD) system, allowing for the use of fast Poisson solvers such as PCG.

Besides using $\tilde{\rho}$ in equations (43) and (44) to enforce the balance condition of equation (38) we also desire a unique velocity at the shared face location $x_{2.5}$ (shown in black in Figure 7) for computing the term $\nabla \cdot \vec{v}^*$ in equation (43) and the term $\nabla \cdot \hat{u}^*$ in equation (44). Since we have separate velocity fields for compressible and incompressible fluids, the values $\hat{u}_{2.5}^*$ and $v_{2.5}^*$ at the shared face need not be equal, even though theoretically the normal component of the velocity field is supposed to be continuous across the interface. Thus, at the shared face we apply a force λ to the incompressible fluid and an equal and opposite force $-\lambda$ to the compressible fluid so that the resulting time t^{**} velocities at the shared face are equal, i.e.,

$$\hat{\rho}_{2.5}^{n+1} \hat{u}_{2.5}^{**} = \hat{\rho}_{2.5}^{n+1} \hat{u}_{2.5}^* - \lambda \Delta t, \quad \rho_I v_{2.5}^{**} = \rho_I v_{2.5}^* + \lambda \Delta t \quad (45)$$

Setting $\hat{u}_{2.5}^{**} = v_{2.5}^{**}$, and solving for λ gives

$$\hat{u}_{2.5}^{**} = v_{2.5}^{**} = \frac{\rho_I v_{2.5}^* + \hat{\rho}_{2.5}^{n+1} \hat{u}_{2.5}^*}{\rho_I + \hat{\rho}_{2.5}^{n+1}} \quad (46)$$

as our velocity at the shared face (note that since the incompressible density tends to be much bigger than the compressible density, one could also use the incompressible velocity and obtain similar results - we have

tested this numerically). Also note that the compressible face velocity is not an actual degree of freedom since the degrees of freedom for compressible flow lie at cell centers. Hence, we add the momentum $-\lambda\Delta t$ to both the compressible cell 2 and the compressible ghost cell 3, i.e., $(\rho u)_2^{**} = (\rho u)_2^* - \lambda\Delta t$ and $(\rho u)_3^{**} = (\rho u)_3^* - \lambda\Delta t$, so that the first equation of equation (45) is satisfied. This is equivalent in spirit to adding $-\lambda\Delta t$ to cell 2 and then re-extrapolating into ghost cell 3. Finally, after changing the compressible state vector at cell 2, the velocity $\hat{u}_{1.5}^*$ is recomputed to make it consistent with the new state in cell 2. In multiple spatial dimensions, if a compressible cell borders multiple incompressible cells then it would be updated multiple times, leading to the compressible velocity at the shared face not matching the incompressible velocity. We use the updated incompressible velocity $v_{2.5}^{**}$ at the shared face in this case for greater accuracy.

After solving the coupled projection solve of equations (43) and (44), the incompressible velocity at the shared face is updated to time t^{n+1} via

$$v_{2.5}^{n+1} = v_{2.5}^{**} - \frac{\hat{p}_3^{n+1} - \hat{p}_2^{n+1}}{\tilde{\rho}\Delta x} \quad (47)$$

For updating the compressible momentum at cell 2, we compute the face pressure $\hat{p}_{2.5}^{n+1}$ using the linear interpolant through p_2^{n+1} and p_{int} , i.e., $p_{2.5}^{n+1} = p_2^{n+1} + (p_{\text{int}} - p_2^{n+1})/2\theta$. If θ is too small for the denominator, we instead compute the slope using equation (42) and obtain $p_{2.5}^{n+1} = p_2^{n+1} + \rho_2^{n+1}(p_3^{n+1} - p_2^{n+1})/(\Delta\tilde{\rho})$. The compressible momentum at cell 2 is then updated as

$$(\rho\vec{u})_2^{n+1} = (\rho\vec{u})_2^{**} - \frac{\hat{p}_{2.5}^{n+1} - \hat{p}_{1.5}^{n+1}}{\Delta x} \quad (48)$$

noting that $(\rho u)_2^{**}$ includes the momentum update $\lambda\Delta t$. For the energy update, equation (36) is still used with the face pressure $\hat{p}_{2.5}^{n+1}$ as computed above and the face velocity $\hat{u}_{2.5}^{n+1}$ at the shared face computed using $\hat{u}_{2.5}^{**}$ in equation (35). Note that in the presence of surface tension, we modify the Poisson equations for both the compressible cell 2 and the incompressible cell 3 by taking the jump $\sigma\kappa$ into account, as described in Section 2.1 (see also [18, 22]).

3.4. Time step restriction

The size of the overall time step is computed as the minimum of the incompressible and the compressible time steps, i.e.,

$$\Delta t = \alpha \min\{\Delta t_I, \Delta t_C\} \quad (49)$$

where α denotes the CFL number. The incompressible time step Δt_I is computed as described in Section 2.1. The compressible time step Δt_C is computed as described in [19]. In order to prevent Δt_C from becoming infinite for near zero velocities u^n , the term $\nabla p/\rho$ is added which estimates the change in velocity at the end of a time step. Hence, the CFL condition becomes

$$\Delta t_C \left(\frac{|u^n|_{\max} + \frac{|p_x|}{\rho_C} \Delta t_C}{\Delta x} \right) \leq 1 \quad (50)$$

Equation (50) is a quadratic in Δt_C with two solutions

$$\frac{-|u^n|_{\max} - \sqrt{|u^n|_{\max}^2 + 4\frac{|p_x|}{\rho_C} \Delta x}}{2|p_x|/\rho_C} \leq \Delta t_C \leq \frac{-|u^n|_{\max} + \sqrt{|u^n|_{\max}^2 + 4\frac{|p_x|}{\rho_C} \Delta x}}{2|p_x|/\rho_C} \quad (51)$$

Note that the lower limit in equation (51) is non-positive, and as $\Delta t_C \geq 0$, only the upper bound needs to be enforced. As $p_x \rightarrow 0$, both the numerator and the denominator vanish obtaining the typical bound of $\Delta x/|u^n|_{\max}$ which is problematic when $|u^n|_{\max}$ is small. We obtain a more convenient time step restriction

which is not plagued by either small $|p_x|$ or small $|u^n|_{\max}$ by replacing the *second* Δt in equation (50) with the right hand bound from equation (51) to obtain

$$\frac{\Delta t_C}{2} \left(\frac{|u^n|_{\max}}{\Delta x} + \sqrt{\left(\frac{|u^n|_{\max}}{\Delta x} \right)^2 + 4 \frac{|p_x|}{\rho_C \Delta x}} \right) \leq 1 \quad (52)$$

In two spatial dimensions, the following CFL restriction is obtained

$$\frac{\Delta t_C}{2} \left(\frac{|u_1^n|_{\max}}{\Delta x} + \frac{|u_2^n|_{\max}}{\Delta y} + \sqrt{\left(\frac{|u_1^n|_{\max}}{\Delta x} + \frac{|u_2^n|_{\max}}{\Delta y} \right)^2 + 4 \frac{|p_x|}{\rho_C \Delta x} + 4 \frac{|p_y|}{\rho_C \Delta y}} \right) \leq 1 \quad (53)$$

3.5. Numerical Results

We used the gamma gas law equation of state $p = (\gamma - 1)\rho e$ and an outer loop of third order TVD Runge-Kutta in all these examples, noting that although the implicit treatment of terms inside the RK loop generally leads to a loss of third order time accuracy, greater stability is achieved. All examples use a CFL number of .5.

3.5.1. One dimensional examples

Consider a computational domain of [0 m, 1 m]. The domain is filled with a compressible gas with $\rho = 1.226 \text{ kg/m}^3$, $u = 0 \text{ m/s}$, and $p = 10^5 \text{ Pa}$. An incompressible droplet of length .2 m is located at the center of the domain with $\rho = 1000 \text{ kg/m}^3$, $u = 100 \text{ m/s}$, and $p = 10^5 \text{ Pa}$. Since the incompressible droplet is moving rightwards in an initially stationary gas, a shock wave forms in the gas ahead of it and a rarefaction wave forms in the gas behind it. Figure 8 shows the density, velocity, pressure and time step profiles, along with the fully explicit method of [5] run on the finest grid of resolution 12800 for the sake of comparison. (Note that our implementation of [5] gave a different result for this example, although agrees with [5] for all other examples that they ran in both one and two spatial dimensions, leading us to believe that there is probably a typo in the description of this example in [5].) Note that our method converges to the highly refined explicit result as the grid is refined. Figure 9 shows similar results when the density of the incompressible droplet is $\rho = 10 \text{ kg/m}^3$. The compressible gas slows down the lighter droplet faster and as a result secondary rarefaction waves stretch between the droplet and the shock and rarefaction waves.

Note that our method can take a time step that is four times larger, however, the cost of each time step is slightly larger because the compressible degrees of freedom have been added to the incompressible Poisson solver. The overall speedup in wall clock time will generally depend on the ratio of increased cost per time step as compared to the decrease in the number of time steps - in this particular example, the code was approximately three times faster. For some problems the speedups can be significantly larger especially when one cares about phenomena that occur after the compressible flow relaxes to smaller velocities - in this case the time steps could increase by several orders of magnitude.

Consider a computational domain of [0 m, 1 m] filled with a compressible fluid with density $\rho = 1.58317 \text{ kg/m}^3$, velocity $u = 0 \text{ m/s}$, and pressure $p = 98066.5 \text{ Pa}$. An incompressible droplet of length .2 m is initially located at the center of the domain with $\rho = 1000 \text{ kg/m}^3$, $u = 0 \text{ m/s}$, and $p = 98066.5 \text{ Pa}$. A shock wave is initially located at $x = .1 \text{ m}$ with a post-shock state of $\rho = 2.124 \text{ kg/m}^3$, $u = 89.981 \text{ m/s}$, and $p = 148407.3 \text{ Pa}$ to the left of $x = .1 \text{ m}$. The shock wave travels to the right impinging on the incompressible droplet, causing both reflected and transmitted waves as shown in Figure 10 at $t = 1.75 \times 10^{-3} \text{ s}$. Note that the transmitted wave is too weak to be seen in this example, although it can be clearly seen in Figure 11 where the incompressible droplet has density $\rho = 10 \text{ kg/m}^3$.

3.5.2. Two dimensional examples

All two-dimensional examples include the effects of viscosity and surface tension with coefficients $\mu = .001137 \text{ kg/ms}$ and $\sigma = .0728 \text{ kg/s}^2$. These effects are not present in the one dimensional examples shown in Section 3.5.1 because the incompressible flow has constant velocity and the interface has no curvature.

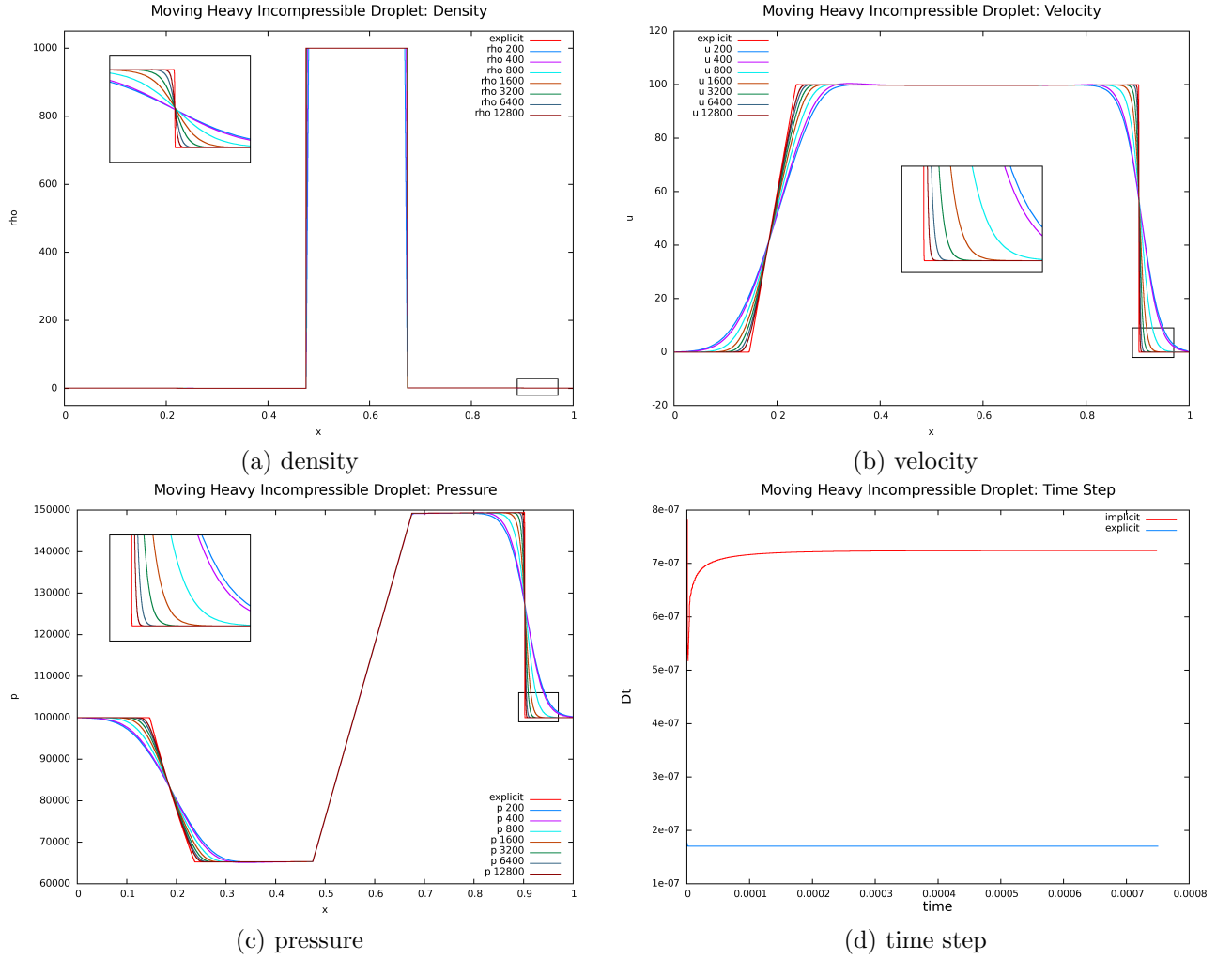


Figure 8: Numerical results for the moving incompressible droplet example, where a droplet of density 1000 kg/m^3 is travelling to the right in an initially stationary compressible fluid at $t = 7.5 \times 10^{-4} \text{ s}$.

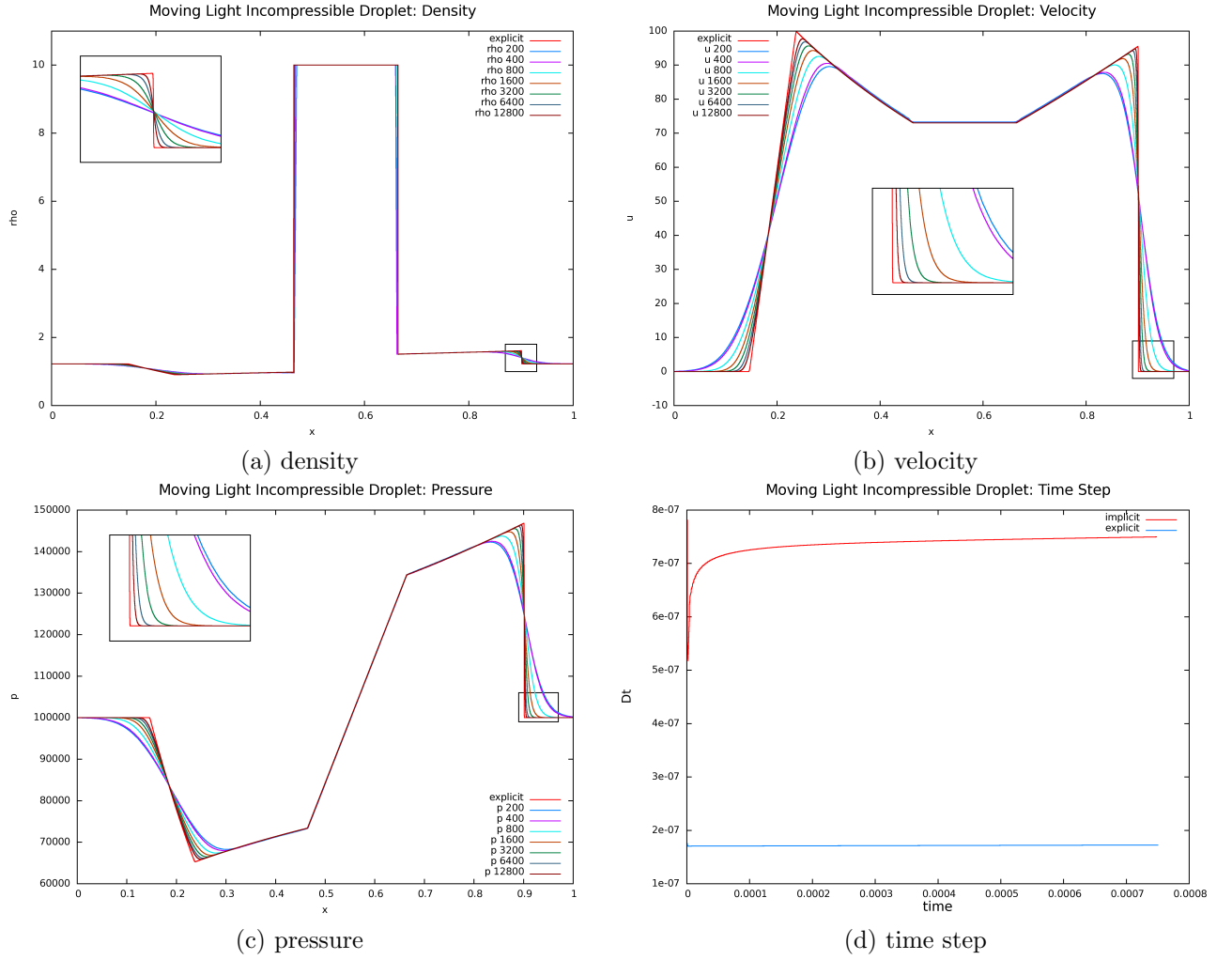


Figure 9: Numerical results for the moving incompressible droplet example, where a droplet of density 10 kg/m³ is travelling to the right in an initially stationary compressible fluid at $t = 7.5 \times 10^{-4}$ s.

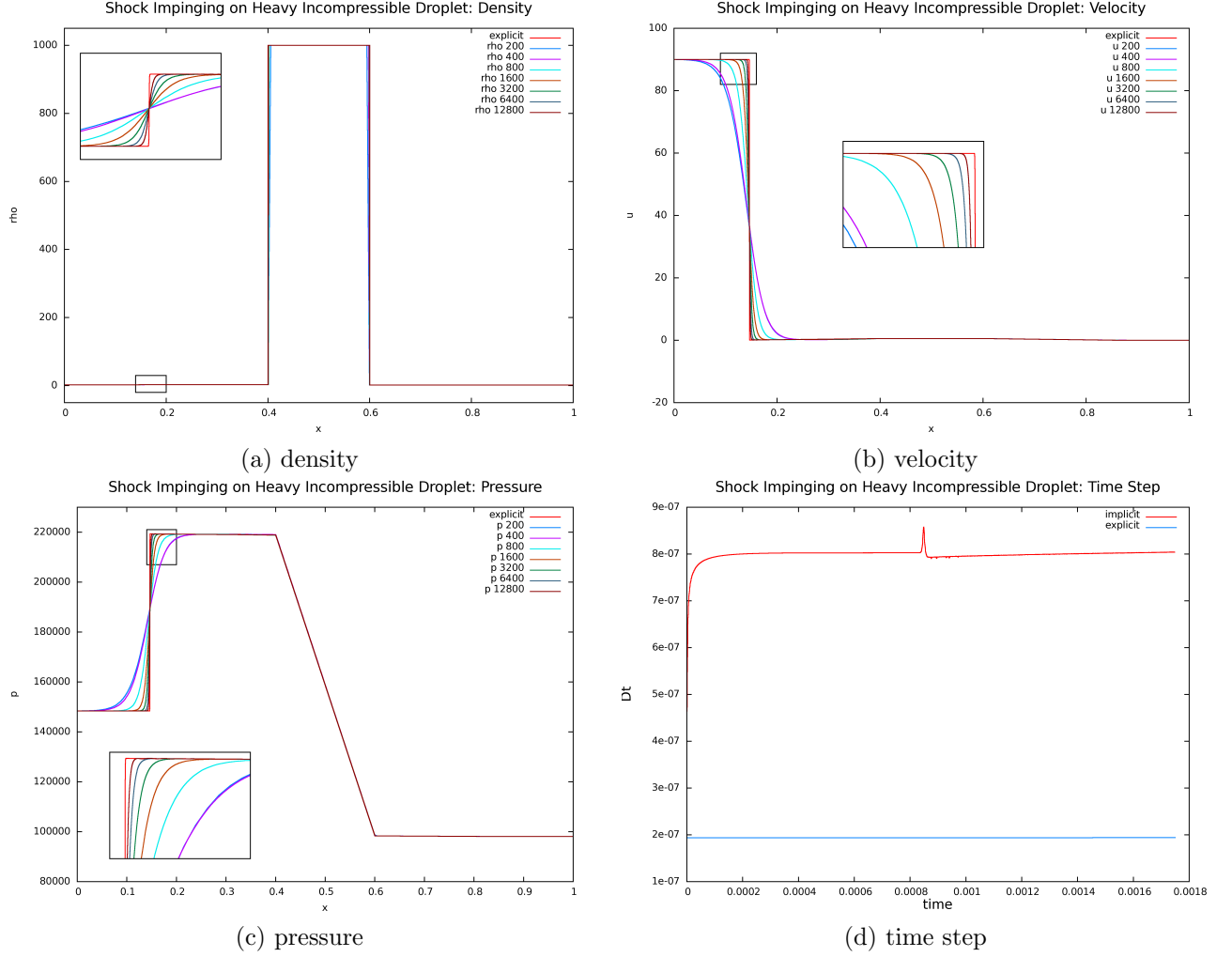


Figure 10: Numerical results for the shock impinging on a heavy incompressible droplet example at $t = 1.75 \times 10^{-3}$ s, where a shock wave initially located at $x = .1$ m travels to the right impinging on an incompressible droplet of density 1000 kg/m^3 generating both reflected and transmitted waves.

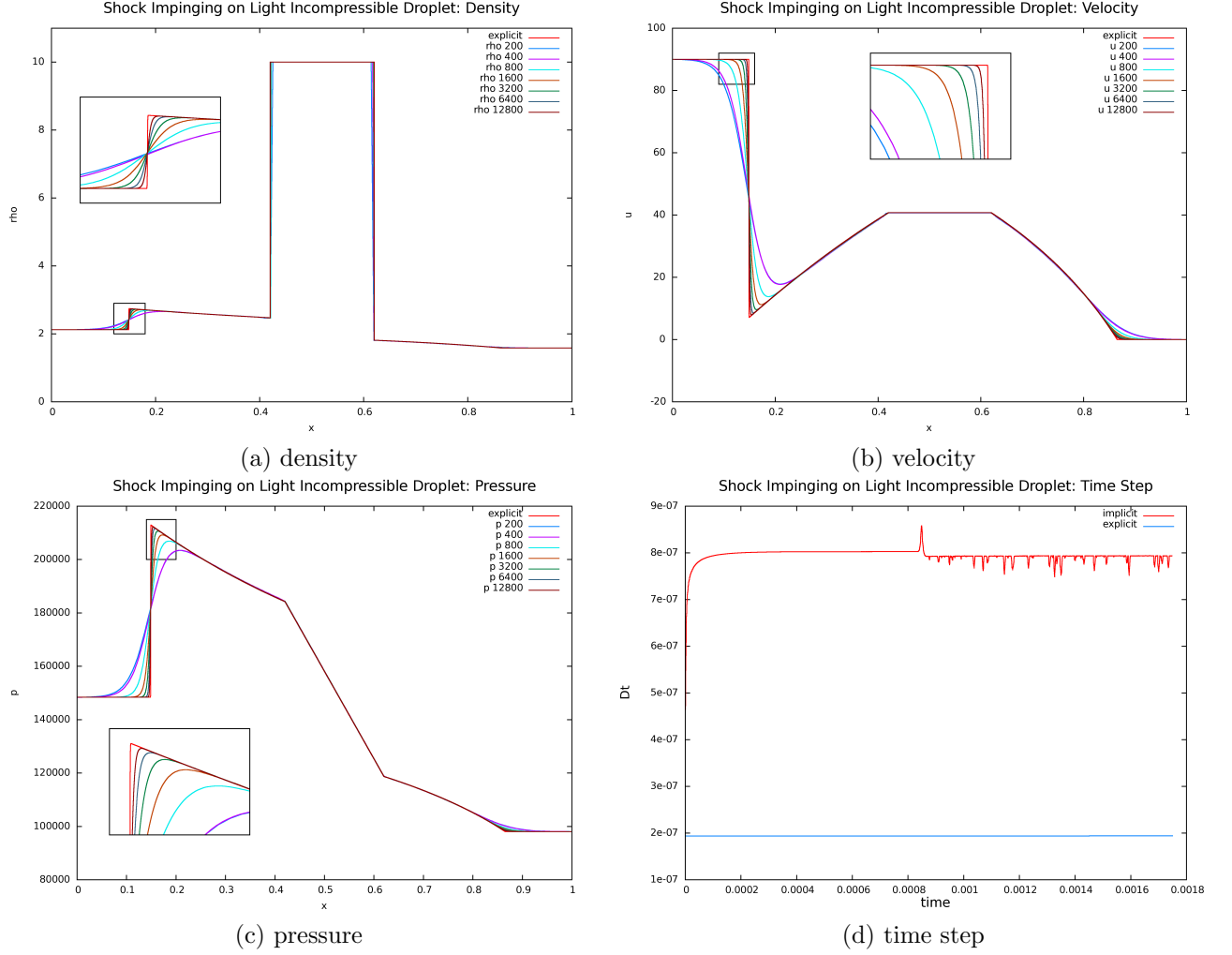


Figure 11: Numerical results for the shock impinging on a light incompressible droplet example at $t = 1.75 \times 10^{-3}$ s, where a shock wave initially located at $x = .1$ m travels to the right impinging on an incompressible droplet of density 10 kg/m^3 generating both reflected and transmitted waves.

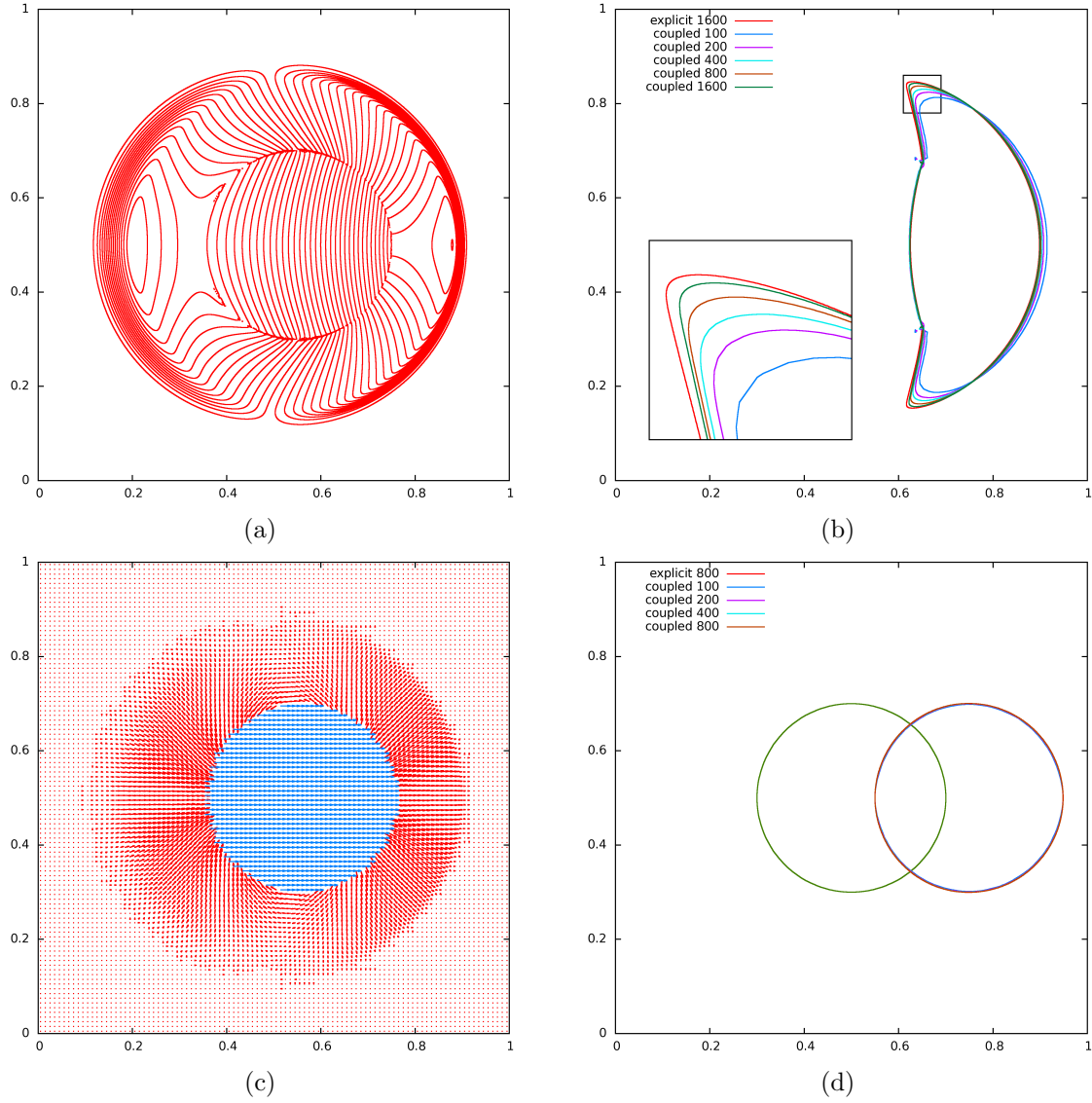


Figure 12: The moving incompressible droplet example in two spatial dimensions, where an incompressible droplet of density 1000 kg/m^3 is travelling to the right in an initially stationary compressible fluid. (a) 50 equally spaced pressure contours between $.75 \times 10^5 \text{ Pa}$ and $1.5 \times 10^5 \text{ Pa}$ on a 1600×1600 grid at $t = 5 \times 10^{-4} \text{ s}$, (b) pressure contour of 1.1×10^5 at $t = 5 \times 10^{-4} \text{ s}$ under grid refinement to illustrate convergence, (c) velocity field at $t = 5 \times 10^{-4} \text{ s}$ where the incompressible velocities are shown in blue, and compressible velocities are shown in red, and (d) the zero level set under grid refinement at $t = 2.5 \times 10^{-3} \text{ s}$ as compared to its initial location.

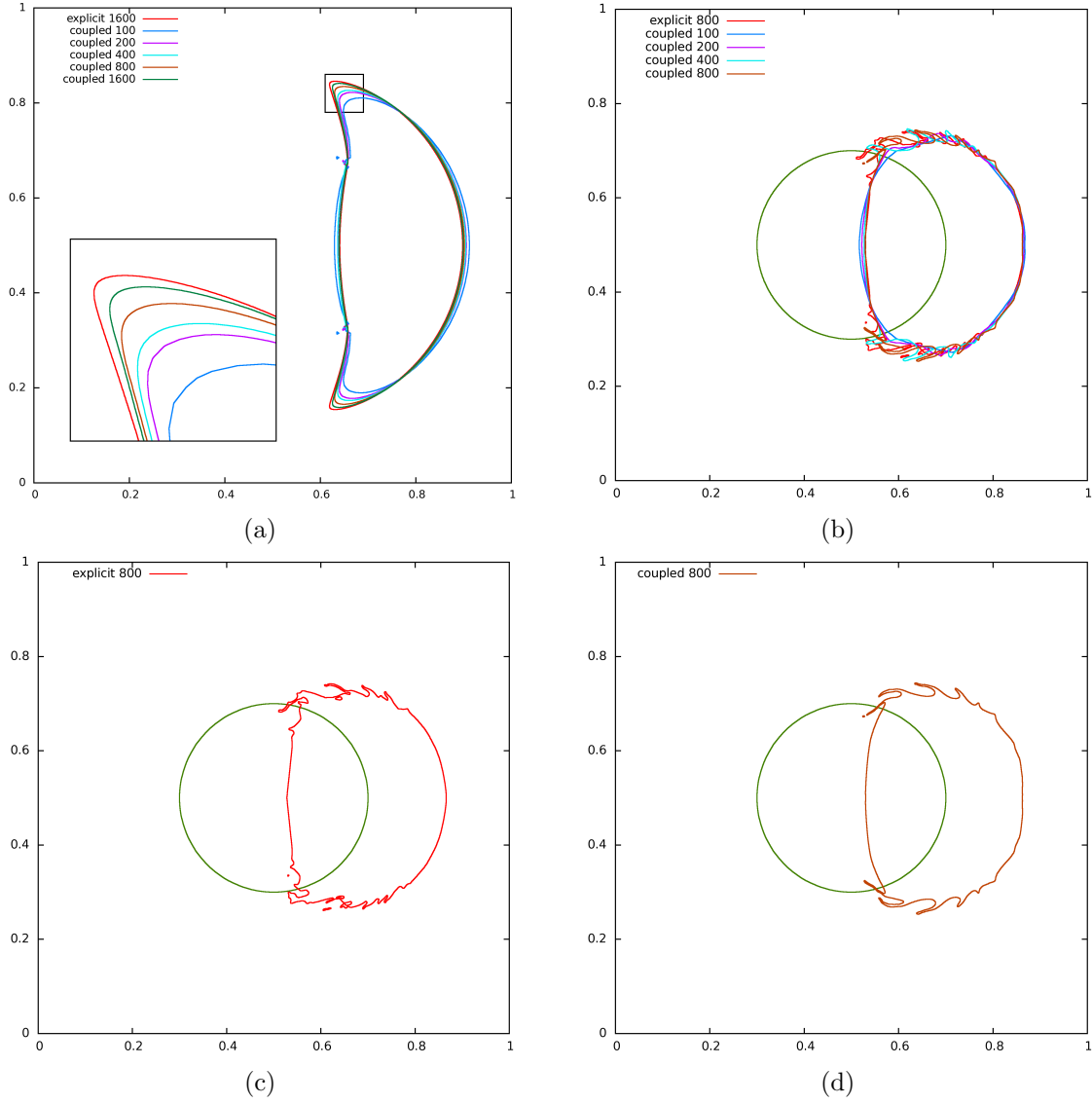


Figure 13: The moving incompressible droplet example in two spatial dimensions, where an incompressible droplet of density 10 kg/m^3 is travelling to the right in an initially stationary compressible fluid. (a) pressure contour of 1.1×10^5 at $t = 5 \times 10^{-4}$ s under grid refinement to illustrate convergence, (b) the zero level set under grid refinement at $t = 2.5 \times 10^{-3}$ s as compared to its initial location, (c) the zero level set on a grid of resolution 800×800 at $t = 2.5 \times 10^{-3}$ s as compared to its initial location using the fully explicit partitioned solver of [5], and (d) the zero level set on a grid of resolution 800×800 at $t = 2.5 \times 10^{-3}$ s as compared to its initial location using our monolithic solver.

Consider a computational domain of $[0 \text{ m}, 1 \text{ m}] \times [0 \text{ m}, 1 \text{ m}]$. Similar to the one dimensional case, the domain is filled with a compressible gas with $\rho = 1.226 \text{ kg/m}^3$, $u = v = 0 \text{ m/s}$, and $p = 10^5 \text{ Pa}$. An incompressible droplet of radius $.2 \text{ m}$ is located at the center of the domain with $\rho = 1000 \text{ kg/m}^3$, $u = 100 \text{ m/s}$, $v = 0 \text{ m/s}$, and $p = 10^5 \text{ Pa}$. Since the compressible gas is initially stationary and the droplet is moving rightwards, a shock wave forms in the gas in front of it, and a rarefaction wave forms in the gas behind it. Figure 12(a) shows 50 equally spaced pressure contours between $.75 \times 10^5 \text{ Pa}$ and $1.5 \times 10^5 \text{ Pa}$ on a 1600×1600 grid at $t = 5 \times 10^{-4} \text{ s}$. Figure 12(b) shows the pressure contour of 1.1×10^5 at various grid resolutions to show that the numerical profiles generated using our method converge to those generated using the fully explicit method of [5] under grid refinement. The velocity field is shown in Figure 12(c) where the incompressible velocities are shown in blue and the compressible ones are shown in red. Figure 12(d) shows the initial location of the zero level set as compared to its location at $t = 2.5 \times 10^{-3} \text{ s}$. Figure 13(a)-(d) show the results for the case when the incompressible droplet has $\rho = 10 \text{ kg/m}^3$. Note that the lighter droplet undergoes larger deformation and also slows down at a faster rate. Also note that, as observed in [5], the computations for the lighter droplet show signs of Kelvin-Helmholtz instability as is apparent by the wiggles in the interface location shown in Figure 13(b). This effect is less apparent on coarser grids because of the artificial damping due to numerical viscosity.

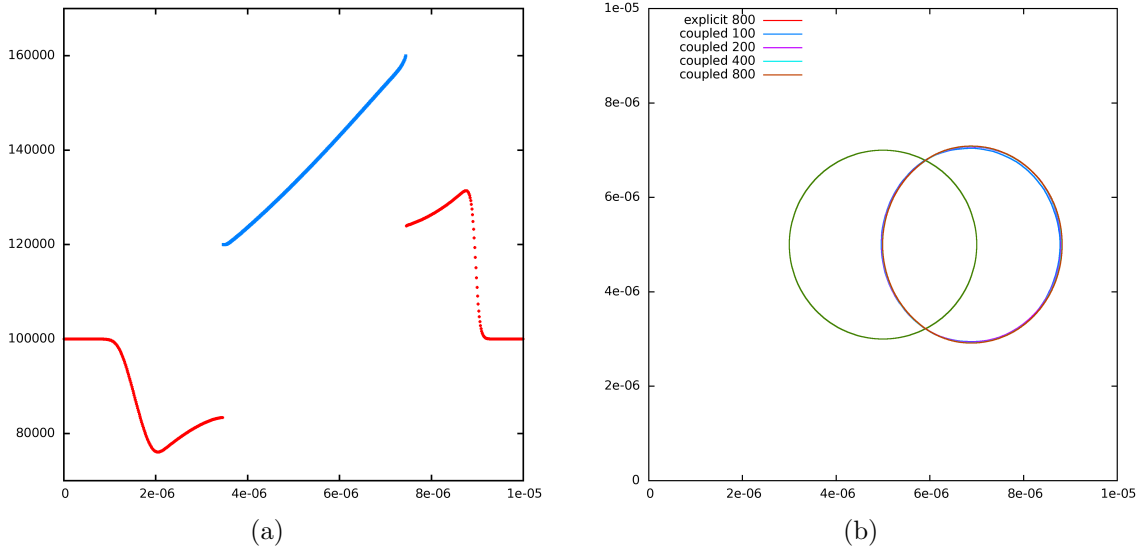


Figure 14: The moving incompressible droplet example in two spatial dimensions, where an incompressible droplet of density 10 kg/m^3 is travelling to the right in a small domain in an initially stationary compressible fluid. (a) one dimensional cross-section of the pressure on an 800×800 grid at $t = 5 \times 10^{-9} \text{ s}$, where the pressure in the incompressible region is shown in blue and that in the compressible region is shown in red, and (b) the zero level set under grid refinement at $t = 2.5 \times 10^{-8} \text{ s}$ as compared to its initial location.

To demonstrate the effects of surface tension and viscosity, we also shrunk the domain to $[0 \text{ m}, 1 \times 10^{-5} \text{ m}] \times [0 \text{ m}, 1 \times 10^{-5} \text{ m}]$ for the case when the incompressible droplet has density $\rho = 10 \text{ kg/m}^3$. Figure 14(a) shows a one dimensional cross-section of the pressure at $t = 5 \times 10^{-9} \text{ s}$, where the pressure in the incompressible region is shown in blue and that in the compressible region is shown in red. Note the jump in pressure across the interface due to surface tension effects. Figure 14(b) shows the initial location of the zero level set as compared to its location at $t = 2.5 \times 10^{-8} \text{ s}$. Note that the smaller droplet is deformed less and has a more spherical shape as compared to the larger droplet shown in Figure 13(d).

Consider a computational domain of $[0 \text{ m}, 1 \text{ m}] \times [0 \text{ m}, 1 \text{ m}]$. Similar to the one dimensional case, the domain is filled with a compressible fluid with $\rho = 1.58317 \text{ kg/m}^3$, $u = v = 0 \text{ m/s}$, and $p = 98066.5 \text{ Pa}$. An incompressible droplet with initial state $\rho = 10 \text{ kg/m}^3$, $u = v = 0 \text{ m/s}$, and $p = 98066.5 \text{ Pa}$ with radius $.2 \text{ m}$ is located at the center of the domain. A shock wave is initially located at $x = .1 \text{ m}$ with a post-shock state

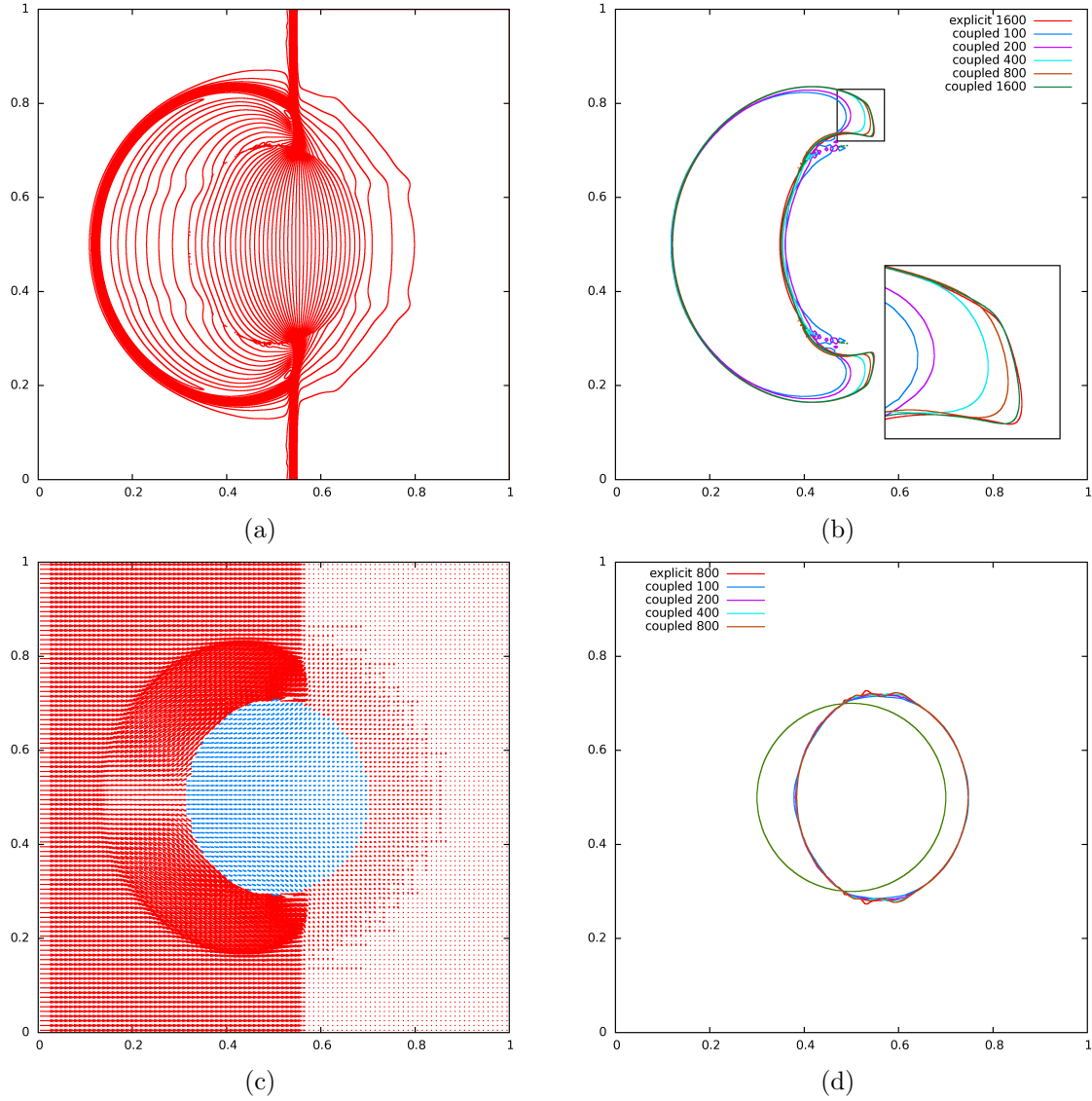


Figure 15: The shock impinging on an incompressible droplet example in two spatial dimensions, where a shock wave initially located at $x = .1$ m travels to the right impinging on an incompressible droplet with $\rho = 10 \text{ kg/m}^3$ generating both reflected and transmitted waves. (a) 50 equally spaced contours between 1×10^5 Pa and 1.8×10^5 Pa on a 1600×1600 grid at $t = 1.25 \times 10^{-3}$ s, (b) pressure contour of 1.62×10^5 Pa under grid refinement at $t = 1.25 \times 10^{-3}$ s to illustrate convergence, (c) velocity field at $t = 1.25 \times 10^{-3}$ s where the incompressible velocities are shown in blue, and the compressible velocities are shown in red, and (d) the zero level set under grid refinement at $t = 2.5 \times 10^{-3}$ s as compared to its initial location.

of $\rho = 2.124 \text{ kg/m}^3$, $u = 89.981 \text{ m/s}$, $v = 0 \text{ m/s}$, and $p = 148407.3 \text{ Pa}$ to the left of $x = .1 \text{ m}$. The shock wave travels to the right impinging on the incompressible droplet, generating both reflected and transmitted waves. Figure 15(a) shows 50 equally spaced pressure contours between $1 \times 10^5 \text{ Pa}$ and $1.8 \times 10^5 \text{ Pa}$ on a 1600×1600 grid at $t = 1.25 \times 10^{-3} \text{ s}$. Figure 15(b) shows the pressure contour of $1.62 \times 10^5 \text{ Pa}$ under grid refinement to illustrate convergence to those generated using the fully explicit method of [5]. Figure 15(c) shows the velocity field, where the incompressible velocities are shown in blue, and the compressible velocities are shown in red. Figure 15(d) shows the initial location of the zero level set as compared to its location at $t = 2.5 \times 10^{-3} \text{ s}$ under grid refinement. Note that the computations on the finer grids also show signs of Kelvin-Helmholtz instability.

3.6. Constant temperature formulation

We make an isothermal assumption, where the equations for conservation of mass and momentum form a closed system and the equation for conservation of energy decouples. In this case, the gamma gas law equation of state can be rewritten as $p = B\rho$, where B is essentially $(\gamma - 1)e$ which we set equal to the atmospheric pressure, i.e., $B = p_{\text{atm}}/\rho_{\text{atm}} \text{ m}^3/\text{kg}$, as described in Section 2. For this equation of state, substituting equation (19), equation (31) becomes

$$p^{n+1} - \Delta t^2 B \rho^n \nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = B \rho^* - \Delta t B \rho^n \nabla \cdot \hat{u}^* \quad (54)$$

where p^* has been replaced with $B\rho^*$. Note that, as mentioned in Section 3.1, $\rho^* = \rho^{n+1}$, and thus $p^* = B\rho^* = B\rho^{n+1}$. Furthermore, if ρc^2 in equation (31) is taken to be at time t^{n+1} instead of t^n , we can replace ρ^n with ρ^{n+1} in equation (54) and divide by $B\rho^{n+1}$ to obtain

$$\frac{1}{\Delta t^2 B \rho^{n+1}} \hat{p}^{n+1} - \nabla \cdot \left(\frac{\nabla \hat{p}^{n+1}}{\rho^{n+1}} \right) = \frac{1}{\Delta t} - \nabla \cdot \hat{u}^* \quad (55)$$

which is the continuous analog of equation (32) for this equation of state.

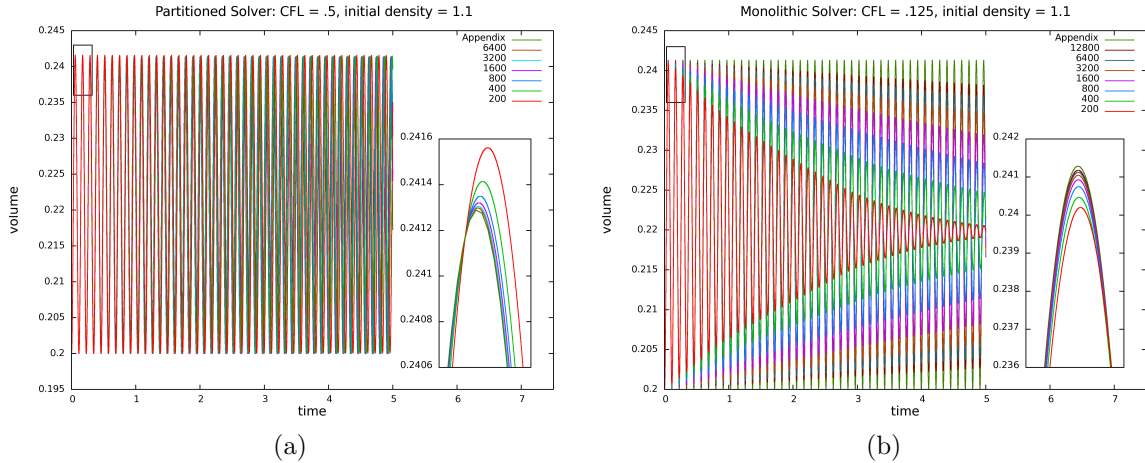


Figure 16: Numerical profiles for bubble volume over time for the oscillating bubble problem in one spatial dimension with equation of state $p = B\rho$ generated by (a) the method of [5], and (b) the monolithic solver.

3.6.1. Numerical results

Now reconsider the examples presented in Section 3.5.1 simulated using $p = B\rho$ as the equation of state. We use the same ambient conditions for density and velocity in all the examples, noting that the ambient

pressures will be different since pressure depends on density. Also, for the examples with prescribed shocks we choose to match the shock speed prescribing a post-shock state of $\rho = 1.97705 \text{ kg/m}^3$ and $u = 70.4023 \text{ m/s}$. Figures 18-21 show the numerical profiles generated using equation (55) - a high resolution comparison to the fully explicit method of [5] is also shown in the results.

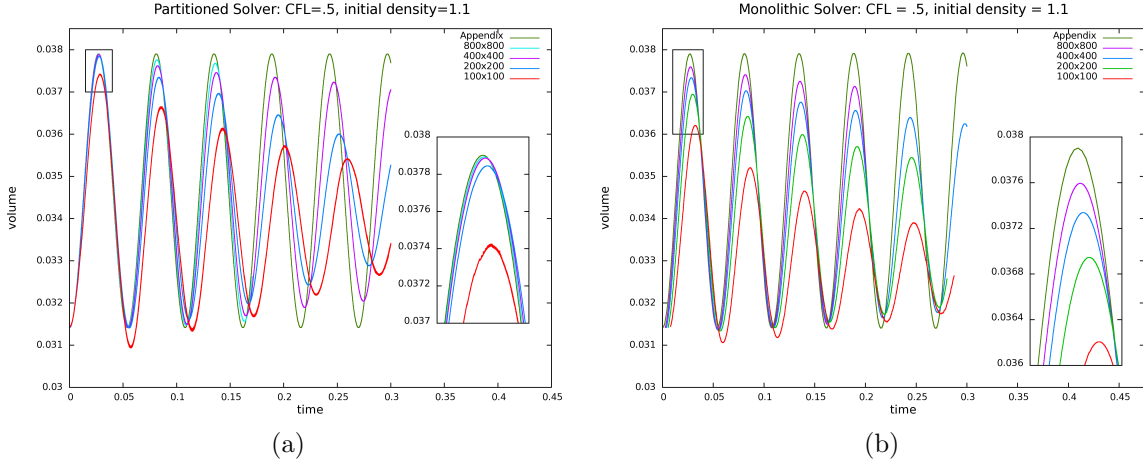


Figure 17: Numerical profiles for bubble volume over time for the oscillating bubble problem in two spatial dimensions with equation of state $p = B\rho$ generated by (a) the method of [5], and (b) the monolithic solver.

Next consider the oscillating bubble problems introduced in Section 2.2. Here we use the two-way coupled simulation techniques proposed in this section which couple the incompressible flow solver to a full compressible flow solver that includes shocks and rarefactions, albeit a somewhat simplified isothermal $p = B\rho$ equation of state. Figure 16(a) shows the numerical profiles for the bubble volume over time generated using the partitioned method of [5], while Figure 16(b) shows the profiles generated using our proposed monolithic solver in one spatial dimension. Note that the results converge to the “exact” solution under grid refinement in both cases. Figure 17 shows the results for a two dimensional oscillating bubble. Note that, unlike the one dimensional case, the method of [5] also damps the solution on coarser grids, although converging to the “exact” solution under grid refinement.

4. Constant density and constant pressure

In this section continuing on from the isothermal assumption from Section 3.6, we additionally make constant density and constant pressure assumptions to arrive at our final method for simulating bubbles.

4.1. Constant density

We achieve constant density by redistributing the density in each bubble as the average density per bubble before the implicit pressure solve, exactly as is done for the partitioned solver in Section 2.3. Note that a spatially constant density field does not imply that pressure inside the bubbles is spatially constant as well. Figure 22 shows the numerical profiles for the bubble volume over time with increasing grid resolutions for the oscillating bubble problems. Note that the profiles converge to the “exact” solutions under grid refinement.

4.2. Constant pressure

We now present a simplified monolithic solver which solves for a constant pressure p^{n+1} inside the bubble with a single degree of freedom, i.e., spatially constant but time-varying. In matrix terms, this corresponds to taking the Poisson matrix for the full system and collapsing all the rows and columns corresponding to

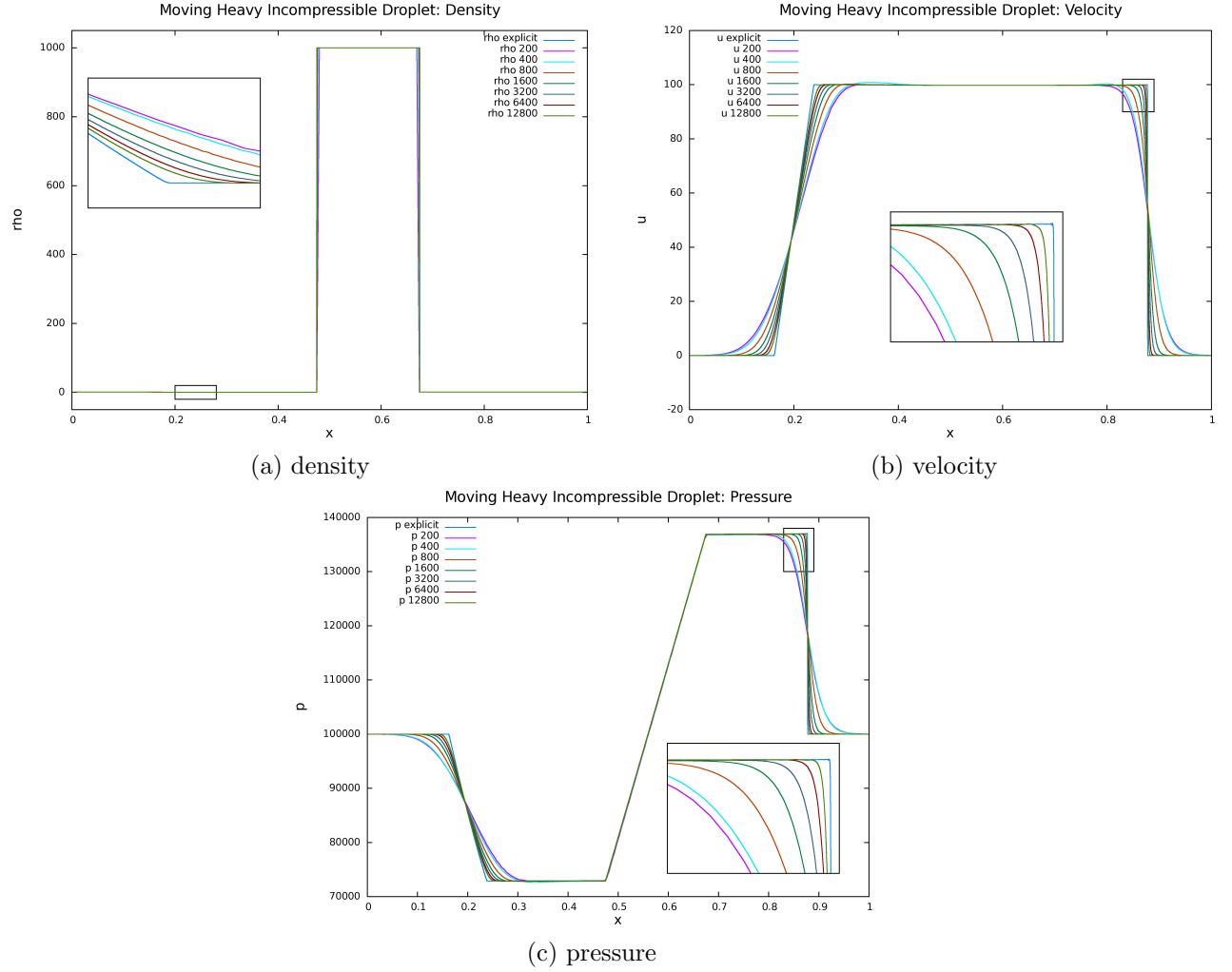


Figure 18: Numerical results for the moving incompressible droplet example, where a droplet of density 1000 kg/m^3 is travelling to the right in an initially stationary compressible fluid with equation of state $p = B\rho$ at $t = 7.5 \times 10^{-4} \text{ s}$.

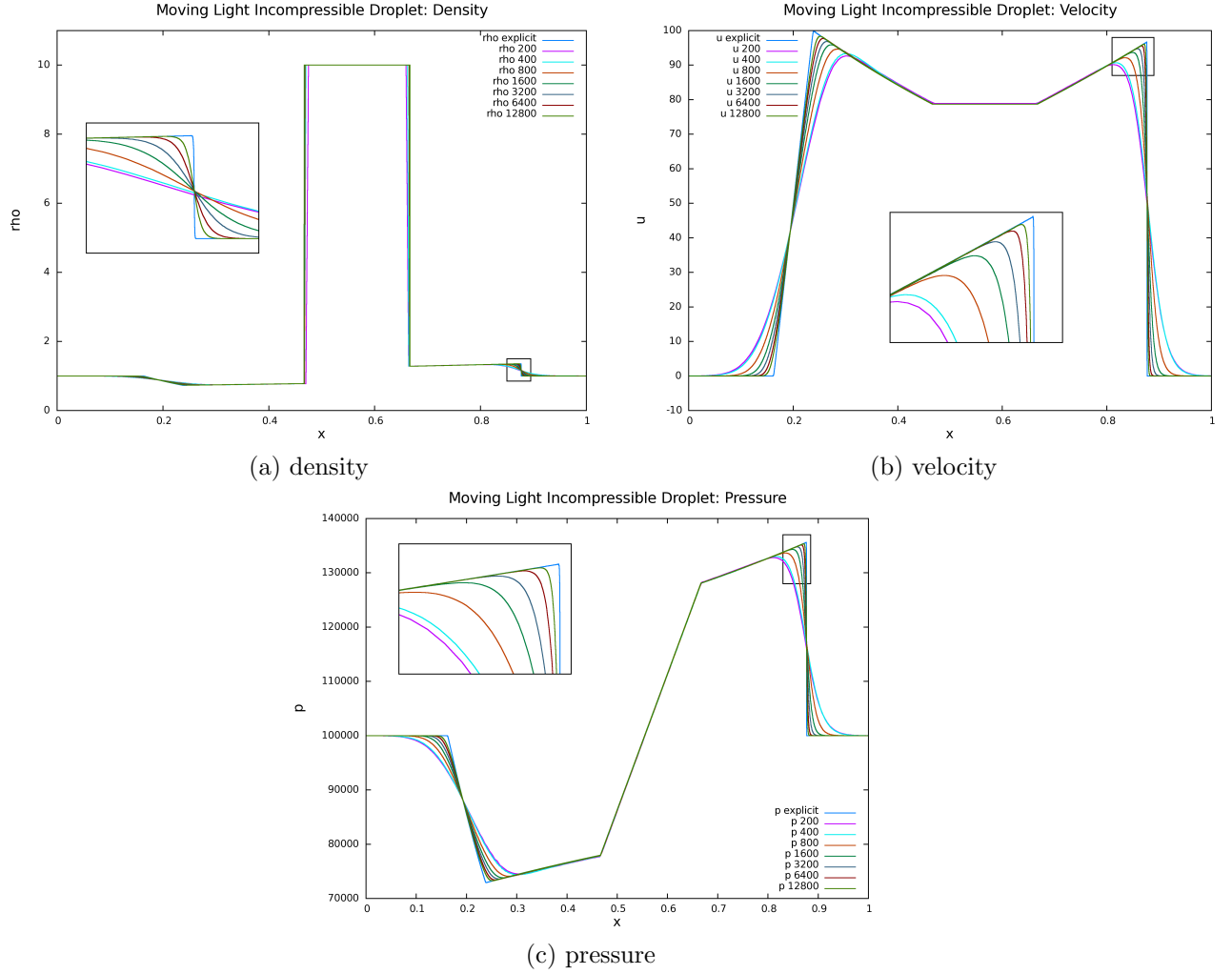


Figure 19: Numerical results for the moving incompressible droplet example, where a droplet of density 10 kg/m^3 is travelling to the right in an initially stationary compressible fluid with equation of state $p = B\rho$ at $t = 7.5 \times 10^{-4} \text{ s}$.

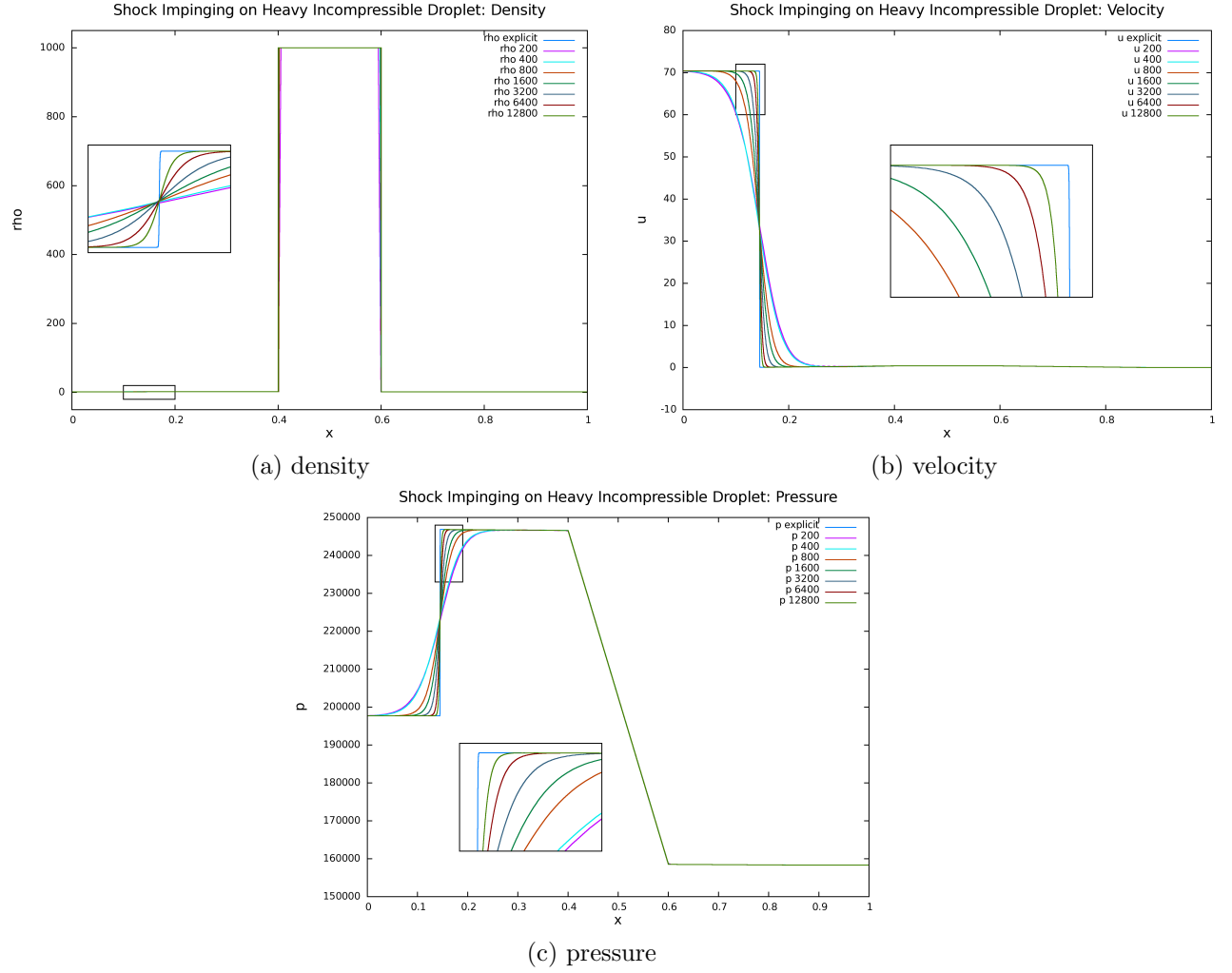


Figure 20: Numerical results for the shock impinging on a heavy incompressible droplet example with equation of state $p = B\rho$ at $t = 1.75 \times 10^{-3}$ s, where a shock wave initially located at $x = .1$ m travels to the right impinging on an incompressible droplet of density 1000 kg/m^3 generating both reflected and transmitted waves.

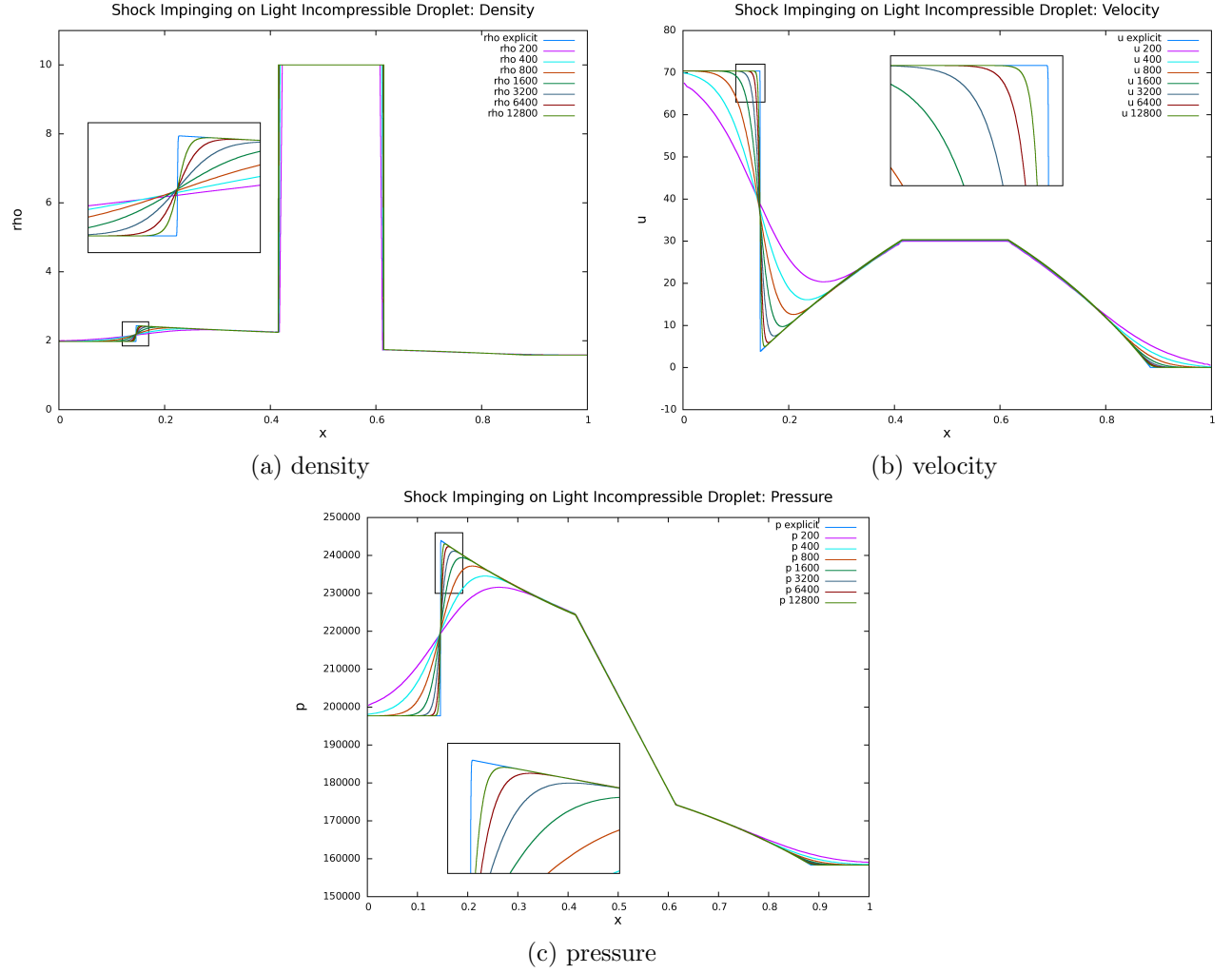


Figure 21: Numerical results for the shock impinging on a heavy incompressible droplet example with equation of state $p = B\rho$ at $t = 1.75 \times 10^{-3}$ s, where a shock wave initially located at $x = .1$ m travels to the right impinging on an incompressible droplet of density 10 kg/m^3 generating both reflected and transmitted waves.

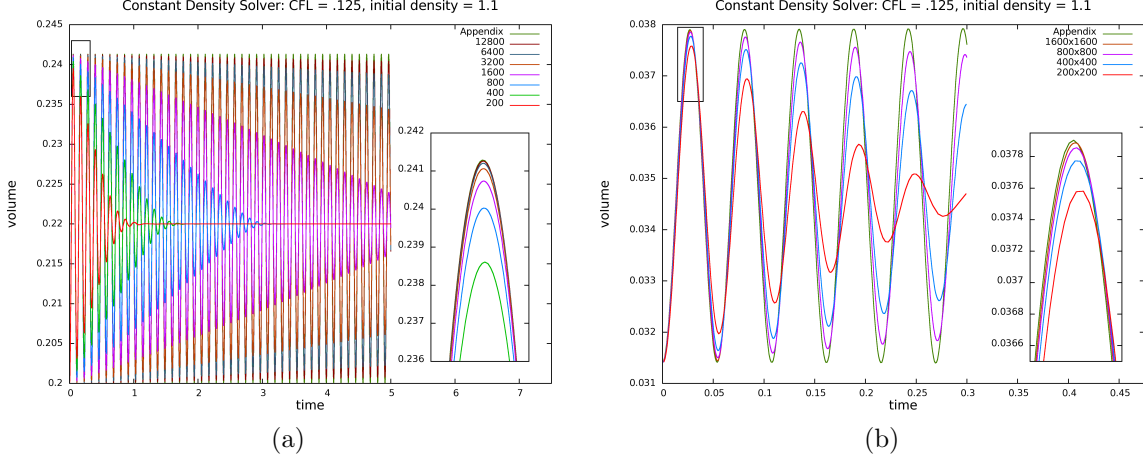


Figure 22: Numerical profiles for bubble volume over time under grid refinement generated using the monolithic solver with density redistribution to obtain constant density as well as an equation of state $p = B\rho$ for the oscillating bubble problem in (a) one spatial dimension, and (b) two spatial dimensions.

cells in the same bubble into a single row and column, adding overlapping matrix elements. This can also be seen as summing equation (55) over the entire bubble to obtain

$$\sum_{i \in \Omega} \frac{1}{\Delta t^2 B \rho^{n+1}} \hat{p}^{n+1} - \sum_{i \in \Omega} \nabla \cdot \left(\frac{\nabla \hat{p}^{n+1}}{\rho^{n+1}} \right) = \sum_{i \in \Omega} \frac{1}{\Delta t} - \sum_{i \in \Omega} \nabla \cdot \hat{u}^* \quad (56)$$

The first term on each side of the equality simply sums over the number of cells N inside the bubble. The last term can be modified by multiplying and dividing by the volume of a cell V_c and converting the volume sum to a surface sum along the MAC grid cell faces that border the bubble. If the average normal velocity on all these faces is \bar{u} and the perimeter of all these faces is \mathcal{P} , we obtain

$$\frac{N}{\Delta t^2 B \rho^{n+1}} \hat{p}^{n+1} - \sum_{i \in \Omega} \nabla \cdot \left(\frac{\nabla \hat{p}^{n+1}}{\rho^{n+1}} \right) = \frac{N}{\Delta t} - \frac{\bar{u} \mathcal{P}}{V_c} \quad (57)$$

Most of the terms in the final summation vanish since ∇p is zero within the bubble, leaving only terms corresponding to MAC grid faces that surround the bubble volume. For each of these faces ρ^{n+1} equals $\bar{\rho}$ as defined in equation (41). Note that the first term on each side of equation (57) is based on the number of cells within the bubble and the second term on each side of the equality is based on the number of MAC grid faces surrounding the bubble. Reminder, ρ^{n+1} in the first term denotes the bubble density.

Note that the constant pressure formulation does not produce pressure gradients within the bubble. Thus, the air velocities are updated through a second projection solve as per equations (15) and (16) using the boundary incompressible velocities, as outlined in Section 2.3. Figures 23(a) and (b) show the numerical profiles for the bubble volume over time for the oscillating bubble problems in one and two spatial dimensions respectively. Note that the profiles closely match those shown in Figures 6 and 22, verifying the correctness of the solver. We also consider the case where the initial density inside the bubble is 1 kg/m^3 , and the outside air pressure is time-varying as $p_{\text{atm}}(t) = Bf(t)$, where $f(t) = 1 - .2 \sin(2\pi t + \pi/2)$. We refer to this problem as the *external pressure field* problem. Figures 23(c) and (d) show the resulting numerical profiles for the bubble volume over time generated using our method in one and two spatial dimensions respectively, which converge to the “exact” solutions under grid refinement. Note that the “exact” solutions are computed using equations (63), (70) and (77) in the appendix which are also valid for time-varying external pressures.

For the oscillating bubble and external pressure field problems, we also considered the effects of surface tension and viscosity for a range of parameters from high to low, both individually and in combination.

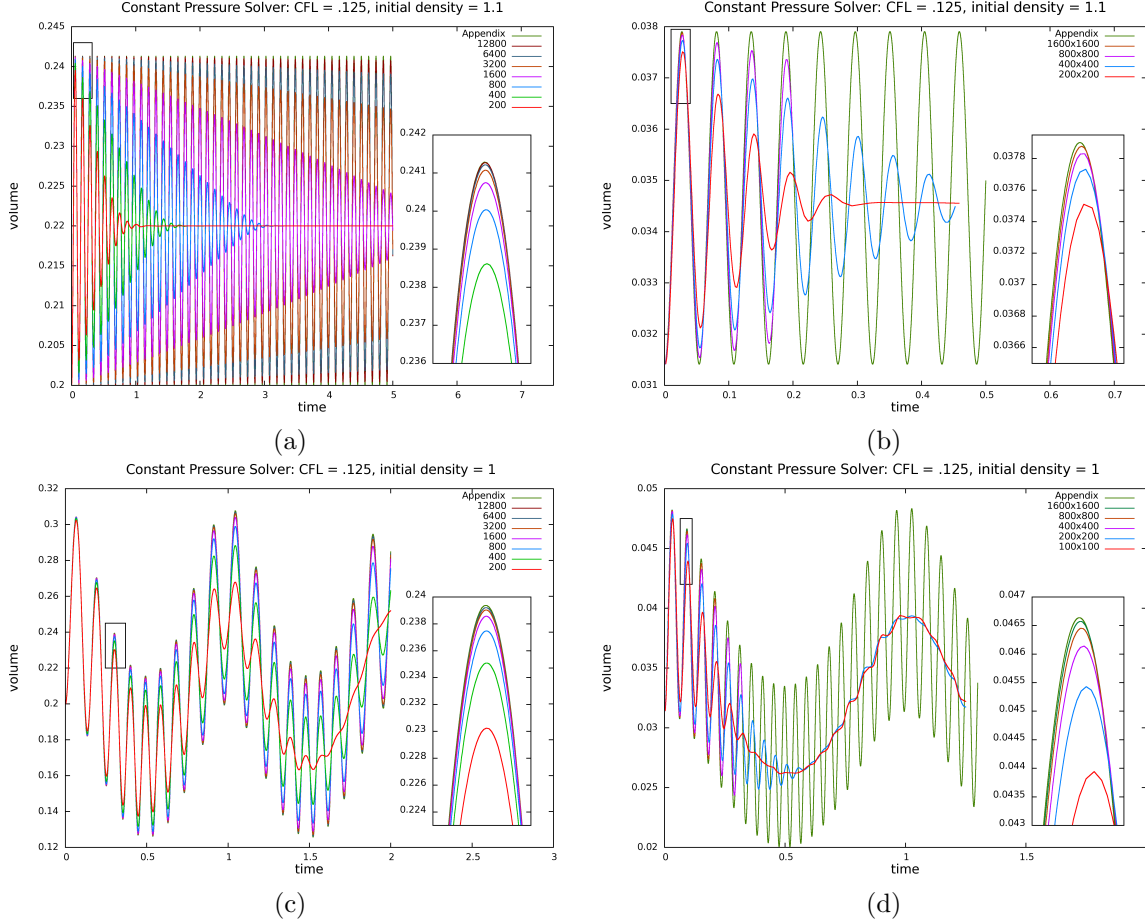


Figure 23: Numerical profiles for the bubble volume over time under grid refinement generated by the constant pressure formulation where all pressure unknowns are collapsed to a single degree of freedom for each distinct bubble for the (a) oscillating bubble problem in one spatial dimension, (b) oscillating bubble problem in two spatial dimensions, (c) external pressure field problem in one spatial dimension, and (d) external pressure field problem in two spatial dimensions.

Figures 24(a) and (b) show the numerical profiles for the bubble volume over time for both these problems in two spatial dimensions with coefficients $\sigma = .0728 \text{ kg/s}^2$ and $\mu = .2 \text{ kg/ms}$. To heighten the effects of surface tension, the computational domain has been uniformly scaled down by 10^{-3} m .

As further validation of our method, we also considered the rising bubble examples from [18]. Consider a computational domain of $[-1 \text{ m}, 1 \text{ m}] \times [-1 \text{ m}, 2 \text{ m}]$ which is initially filled with water except for a circular air bubble of radius $1/3 \text{ m}$ centered at the origin with density $\rho = 1.226 \text{ kg/m}^3$. The effects of surface tension and viscosity are present with coefficients $\mu = .001137 \text{ kg/ms}$ and $\sigma = .0728 \text{ kg/s}^2$. The edges of the computational domain have solid wall boundary conditions. Figure 25 shows the positions of the air bubble at $t = 0$, $t = .2$, $t = .35$ and $t = .5$ seconds under grid refinement. Note that the results are similar to those shown in Figures 3 and 4 from [18]. The finer grid computations at $t = .35 \text{ s}$ and $t = .5 \text{ s}$ show signs of Kelvin-Helmholtz instability, as noted in [18]. To demonstrate the effects of surface tension, we reduced the computational domain to $[-.01 \text{ m}, .01 \text{ m}] \times [-.01 \text{ m}, .02 \text{ m}]$ and the radius of the air bubble to $1/300 \text{ m}$. Figure 26 shows the positions of the air bubble at $t = 0$, $t = .02$, $t = .035$ and $t = .05$ seconds under grid refinement. Note that the results are similar to those shown in Figures 1 and 2 from [18].

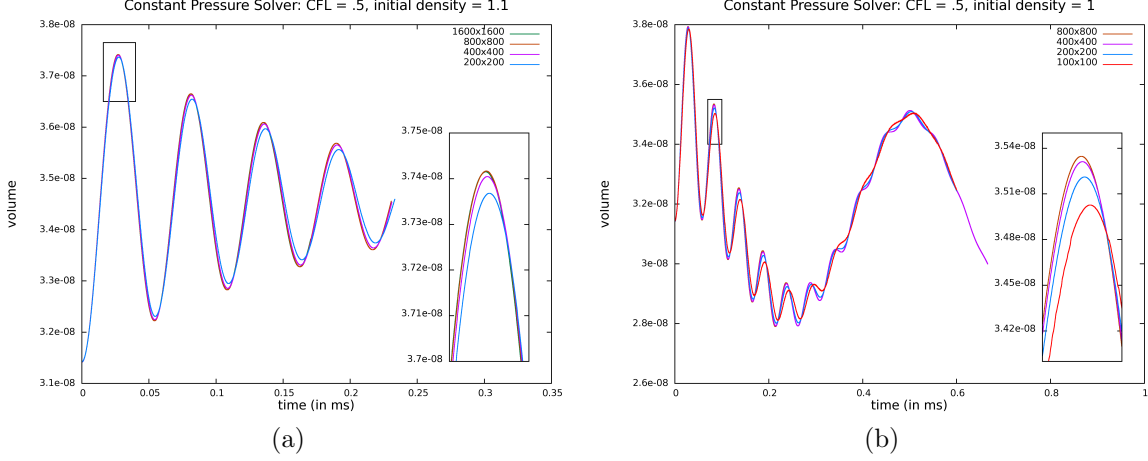


Figure 24: Numerical profiles for the bubble volume over time under grid refinement generated by the constant pressure solver under the effects of surface tension and viscosity for the (a) oscillating bubble problem in two spatial dimensions, and (b) external pressure field problem in two spatial dimensions.

5. Momentum conservation

Consider an isolated incompressible droplet with no ambient pressure forces. When solving equation (4) to make the velocities divergence free, a constant pressure along the boundary leads to a net force of zero implying momentum conservation for the droplet. When updating the velocity degrees of freedom on regular MAC grid faces via equation (5), the interior cell-centered pressures are applied in a conservative fashion to velocities at faces bordering interior cells, and as there is no net pressure force along the boundary the projection step conserves momentum in each Cartesian direction independently. That is, momentum is conserved during the projection step for the velocity degrees of freedom that surround the cell-centered pressure degrees of freedom.

During the velocity extrapolation step, if one computes ϕ -values at faces by averaging cell-centered ϕ -values, then some of the faces involved in the momentum conserving projection step above may be deemed outside the droplet and overwritten. This violates momentum conservation and can be seen as moving the boundary inward by one grid cell replacing the proper exterior constant pressure Dirichlet boundary condition with a spurious internal pressure. Thus we do not use face-averaged ϕ -values, instead labeling every face adjacent to an interior cell center as interior to the droplet. Similarly, the viscous solver uses the same velocity degrees of freedom with exterior Neumann boundary conditions. Currently, we do not use momentum conserving advection [21] as the largest errors for momentum conservation occur when applying surface tension.

In the presence of surface tension, there is additional momentum added for any velocity degree of freedom that lies between an interior and an exterior cell. We compute curvature at a face as a ϕ -weighted average of its cell-centered values, i.e.,

$$\kappa_f = \frac{\kappa_i |\phi_{i+1}| + \kappa_{i+1} |\phi_i|}{|\phi_i| + |\phi_{i+1}|} \quad (58)$$

The surface tension force per unit area at this face is $\text{sign}(\pm 1)\sigma\kappa$, where $\text{sign}(\pm 1)$ is chosen consistent with the outward unit normal in the Cartesian grid directions based on which cells are interior and exterior to the level set. To properly conserve momentum, the net force due to surface tension should sum to zero for each independent bubble and droplet independently along each of the Cartesian directions. To enforce this for each independent bubble and droplet, we compute the total surface tension force per unit area along each Cartesian direction as $\sigma\kappa^{\text{total}} = \sum_f \text{sign}(\pm 1)\sigma\kappa_f$, and subtract it off from the corresponding force for each

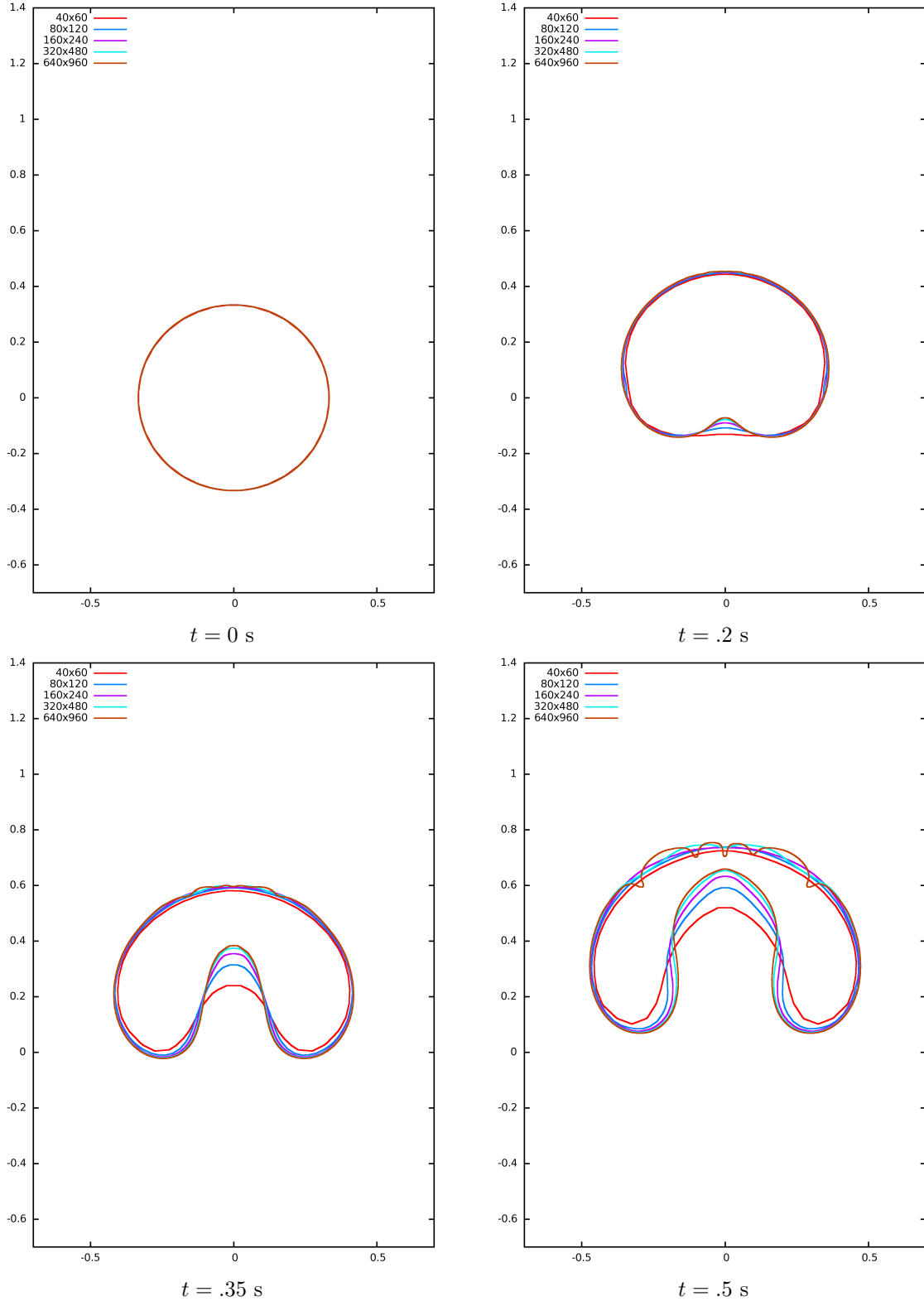


Figure 25: A large air bubble rising in water with solid wall boundary conditions under grid refinement. Note that the computations at time $t = .5$ s show signs of Kelvin-Helmholtz instability.

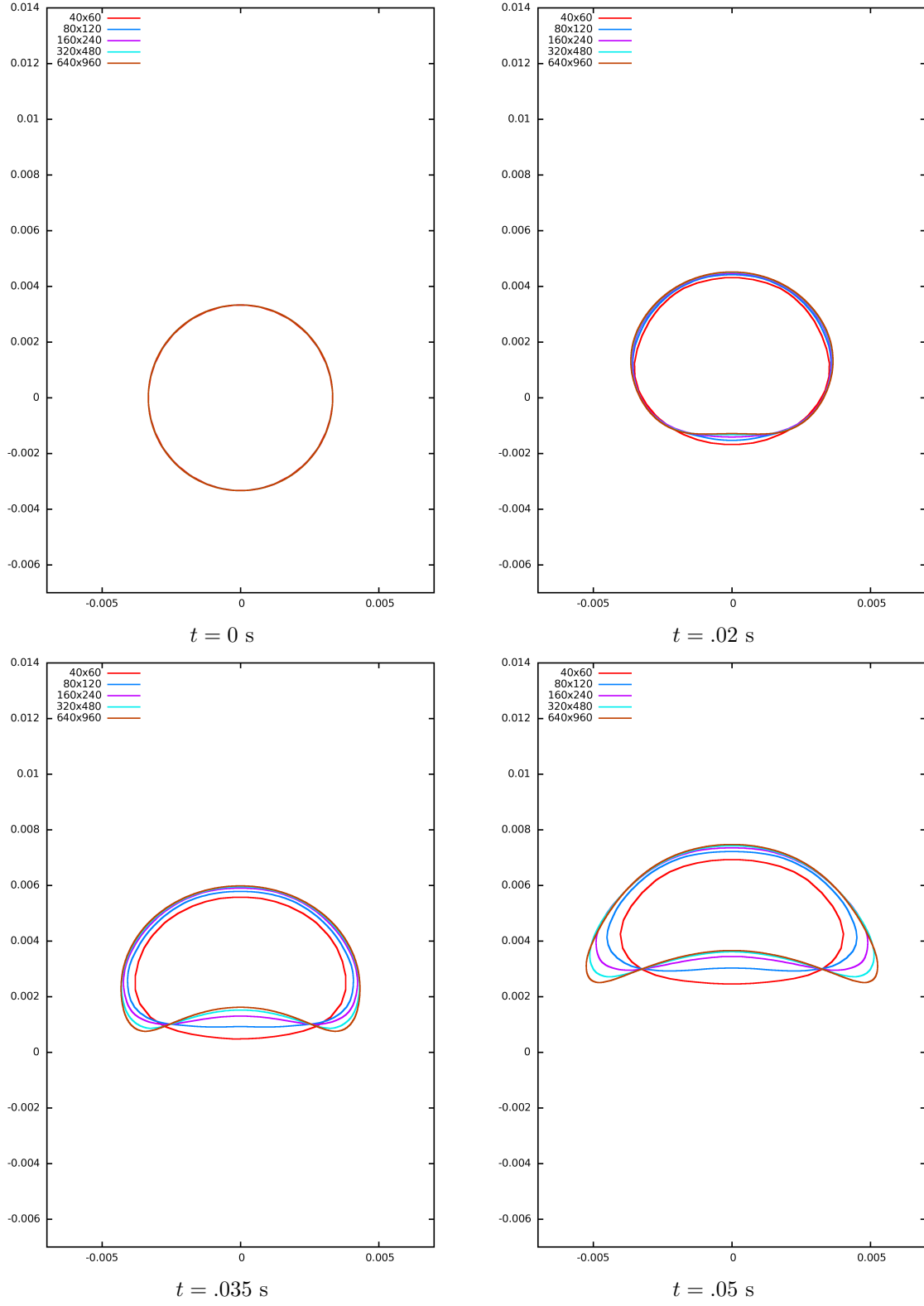


Figure 26: Level set profiles generated using the constant pressure solver for a small air bubble rising in water with solid wall boundary conditions under the effects of surface tension and viscosity.

face in a curvature-weighted fashion. That is, the new jump for each face becomes

$$\sigma\kappa_f^{\text{new}} = \text{sign}(\pm 1)\sigma\kappa_f - \frac{|\kappa_f|}{\sum_f |\kappa_f|} \sigma\kappa^{\text{total}} \quad (59)$$

Obviously alternatives to equation (59) exist, and it is not the form of the correction but the fact that a correction needs to be made in order to conserve momentum that we stress. In our simulations we noticed that the net surface tension force along the boundary is close to zero for large well-resolved bubbles and water droplets. However, for under-resolved droplets which are a few grid cells wide, we noticed that the net surface tension force can be so far from zero that droplets can even change directions in mid-air violating conservation of momentum.

Although we only apply the surface tension correction to closed surfaces, it appears that one can use a similar strategy for open surfaces connected to boundaries as well. Consider for example a single circular bubble, if this bubble is cut in half then it follows from the conservation of momentum that the force on the top half of the bubble must equal the force on the bottom half. However, if we redrew the bottom half of the bubble to any arbitrary curve, then the net force on this new bottom half must still be equal to the net force on the top half. Essentially, the net force on the bottom half of the bubble is independent of its shape and is somehow tied to the boundary conditions on the top half. Therefore, for an open surface connected to the boundary of the domain, with additional knowledge about contact angles, one could likely compute a consistent measure for the net force on the open surface.

6. Complex bubble breakup

To summarize, our method for simulating bubbles in free surface incompressible flows is as follows. We use the level set method [24] to track the interface between the bubbles and the incompressible fluid. Initially, each bubble is assigned a density (or mass) which is advected using the unconditionally stable conservative semi-Lagrangian scheme of [21]. We advect the bubble density using air velocities which are constructed and maintained in a separate velocity field, although one could also use velocities extrapolated from the incompressible flow for increased efficiency but lower accuracy. The velocity used for level set advection is a hybrid between the incompressible flow velocity and the air velocity field. Despite more accurate velocities, numerical smearing and other errors cause the location of the zero level set and the location of the non-zero bubble densities to drift apart over time. We address this issue as follows. First we compute the total mass that belongs to a bubble as the sum of all the mass inside the bubble and all the mass closest to that bubble. Then we use a flood fill algorithm on that bubble to identify all grid cells belonging to that connected component. The volume of this connected component is carefully computed using a piecewise linear reconstruction of the level set as outlined in [20]. The mass is then uniformly redistributed inside the bubble to obtain a spatially constant bubble density. Before advecting the incompressible velocities and the air velocities, they are extrapolated across the interface in order to define ghost node values using the fast extension method of [1]. The two velocity fields are then independently advected using the second order accurate MacCormack method [27]. The level set function is advected using the particle level set method of [7] and the semi-Lagrangian advection scheme of [8]. To keep the level set a signed distance function we use the modified fast marching method of [23]. Note that we also compute the advection terms in a thin band of ghost cells so they are adequately defined when the interface moves. Also note that we passively advect the bubble mass that is not close to any bubble using the incompressible flow velocities in order to accurately track sub-grid level details. When a bubble is about to merge with the ambient air near the free surface, dynamic events can cause the bubble to open in one time step and close in the next. To robustly track the bubble density in these cases, we keep advecting the bubble mass even if the bubble merges with the ambient air near the free surface.

Viscosity in the water is treated implicitly using Neumann boundary conditions at the interface that the derivative of each component of the incompressible velocity is zero. As described in Section 2.1, this is based on the assumption that the dynamics inside the air bubbles contain little momentum and hence, they cannot absorb any viscous momentum from the liquid. Thus, the jump in pressure due to viscosity is also zero since

the normal component of the viscous stress vanishes across the interface. Finally, as we assume constant viscosity in the incompressible fluid, the equations for the different components of the incompressible velocity decouple as in equations (6) and (7) for two spatial dimensions. Note that the level set is advected to its new time t^{n+1} position before the viscous update due to the need to prescribe interface boundary conditions.

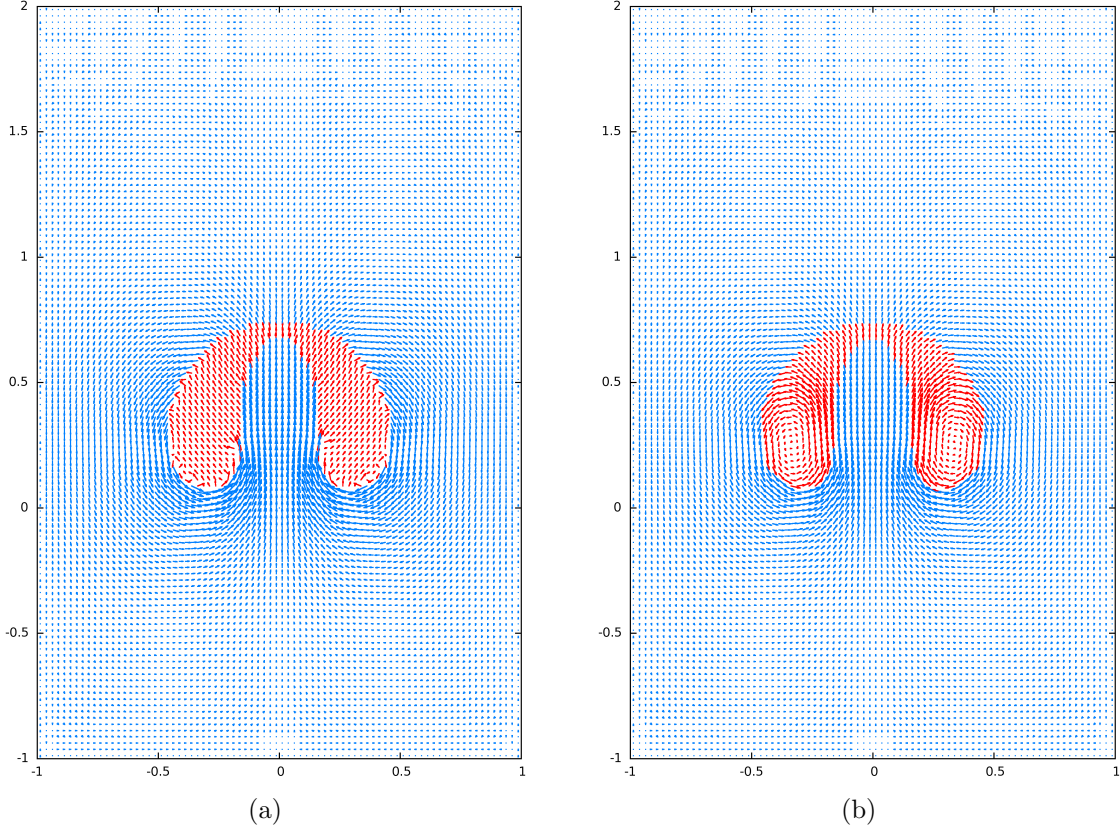


Figure 27: Velocity field at $t = .5$ s on a 80×120 grid for the rising bubble example from [18] (and Figure 25) when the air velocities are (a) wiped out at the end of every time step and computed from the boundary incompressible velocities using a second projection solve, and (b) advected forward in time and updated using pressure gradients from a second projection solve. The incompressible velocities are shown in blue and the air velocities are shown in red. Note that the velocity field in (b) appears much more continuous and smooth compared to the velocity field in (a).

For the implicit step of our method, we use a coupled solve between a single degree of freedom pressure for the bubble and the surrounding incompressible flow by solving equation (57), as described in Section 4.2. The ambient air is taken to be a Dirichlet boundary condition of $p_{\text{atm}} = 101,325$ Pa for which we use the second order cut-cell method of [11]. In the presence of surface tension, the appropriate $\sigma\kappa$ jumps are added to pressure values near the interface. These pressure jumps are carefully computed noting that the net surface tension force on each bubble and each water droplet must be zero, as described in Section 5. We use this pressure to update the incompressible velocities via equation (5). This provides a very stable monolithic coupling for interactions between bubbles and the surrounding incompressible flow. The air velocities inside the bubbles are computed from the boundary incompressible flow velocities using a second projection as in equations (15) and (16). One might consider wiping out the air velocities at the end of every time step assuming that the bubble has little momentum and thus, a very small influence on the incompressible velocity field. However, this destroys the temporal continuity of the overall flow field as shown in Figures 27(a) and (b) for the rising bubble example from Figure 25. Note that all the vorticity in the air velocities is confined near the boundaries in Figure 27(a) which would give rise to a noisy level set as the simulation progresses.

In contrast, Figure 27(b) has a much better behaved velocity field which maintains a smooth level set over time.

Finally, the size of the time step is computed using equation (10) where we add an estimate for $|\nabla p|$ using equation (14) which accounts for the change in velocity over a time step. This term prevents the time step from becoming excessively large when velocities are near zero and is similar in spirit to the idea proposed in [18] for body forces and [19] for compressible flow.

6.1. Numerical results

Consider a computational domain of $[-1 \text{ m}, 1 \text{ m}] \times [-1 \text{ m}, 2 \text{ m}]$ which is initially filled with water with density 1000 kg/m^3 where the free surface is located at $y = 1.5 \text{ m}$. A circular air bubble of radius $1/3 \text{ m}$ is centered at the origin with density 1.364 kg/m^3 . The effects of surface tension and viscosity are absent. Figure 28 shows the level set at $t = 0, t = .45, t = .9, t = 1.2, t = 1.5, t = 1.8, t = 2.4, t = 2.7$ and $t = 3$ seconds for two grids of resolutions 320×480 and 640×960 respectively. Note the small scale details that our solver is able to resolve and accurately track over time. In order to show convergence of our solver under grid refinement, we reduced the computational domain to $[-.01 \text{ m}, .01 \text{ m}] \times [-.01 \text{ m}, .02 \text{ m}]$ and the radius of the bubble to $1/300 \text{ m}$, and included the effects of surface tension and viscosity with coefficients $\sigma = .0728 \text{ kg/s}^2$ and $\mu = .001137 \text{ kg/ms}$. The initial density of the air bubble is 1.227 kg/m^3 . Figure 29 shows the level set at $t = 0, t = .02, t = .04$ and $t = .08$ seconds.

As further validation, we also simulated the rising bubble experiment corresponding to Fig 1A from [16] where a bubble of radius $.0061 \text{ m}$ and initial density 1 kg/m^3 rises in a liquid of density 875.5 kg/m^3 . The computational domain is $[-.1464 \text{ m}, .1464 \text{ m}] \times [-.0732 \text{ m}, .366 \text{ m}]$ and the bubble is initially located at the origin. The coefficient of viscosity is $\mu = .118 \text{ kg/ms}$ and the coefficient of surface tension is $\sigma = .0322 \text{ kg/s}^2$. The edges of the computational domain have slip solid wall boundary conditions. This example was also simulated in [31, 15] and our solver gives similar results. Figure 30(a) shows the bubble at time $t = .6 \text{ s}$ on a 512×768 grid along with the streamlines both inside the bubble and in its wake. Figure 30(b) shows the time evolution of the position of the center of mass of the bubble. We compared this data with the linear best fit for $.6 \leq t \leq .8$ seconds. The slope of the linear best fit is $.2114 \text{ m/s}$ whereas the expected slope is $.215 \text{ m/s}$. We note that the streamlines do not match the experimentally observed values due to the fact that a highly simplified single pressure degree of freedom model is used for the air, however, the steady state bubble rise speed is close to the experimentally measured value in [16] and the steady state bubble shape is similar to that observed in [16] and computed in [31, 15].

6.1.1. Object interaction

When updating the incompressible flow velocities during advection, in the presence of objects, we set velocity Dirichlet boundary conditions at cell faces whose centers lie inside an object as these cell faces have well-defined velocities determined by the object velocity. We then advect every face obtaining a well-defined incompressible flow velocity field at time t^* . As we consider interaction with objects that are fully submerged initially, we initialize ϕ inside these objects to be the value when no objects were present, i.e., treat them as water. Subsequently when advecting ϕ , we also update ϕ at grid cells whose centers lie inside an object. Air velocities are handled similar to the incompressible flow velocities during advection with Dirichlet boundary conditions at cell faces inside objects. As described in [21] (see also [14]), for advecting the air density, we modify the forward and backward ray casting to stop when it hits an object. Interpolation weights are computed at the surface point, where weights coming from cells inside an object are discarded and the remaining weights are rescaled to sum to 1. During the viscous solve, we set Dirichlet boundary conditions at cell centers that lie inside objects. When solving for the pressure to make the incompressible flow velocities divergence free, we set Neumann boundary conditions at cell faces whose centers lie inside objects. Similarly, during the second projection step for updating the air velocities, we set Neumann boundary conditions at cell faces whose centers lie inside objects.

Consider a computational domain of $[-1 \text{ m}, 1 \text{ m}] \times [-1 \text{ m}, 2 \text{ m}]$ which is initially filled with water with density 1000 kg/m^3 where the free surface is located at $y = 1.5 \text{ m}$. A circular air bubble of radius $1/3 \text{ m}$ is centered at the origin with density 1.364 kg/m^3 . The domain has nine circular objects with four objects

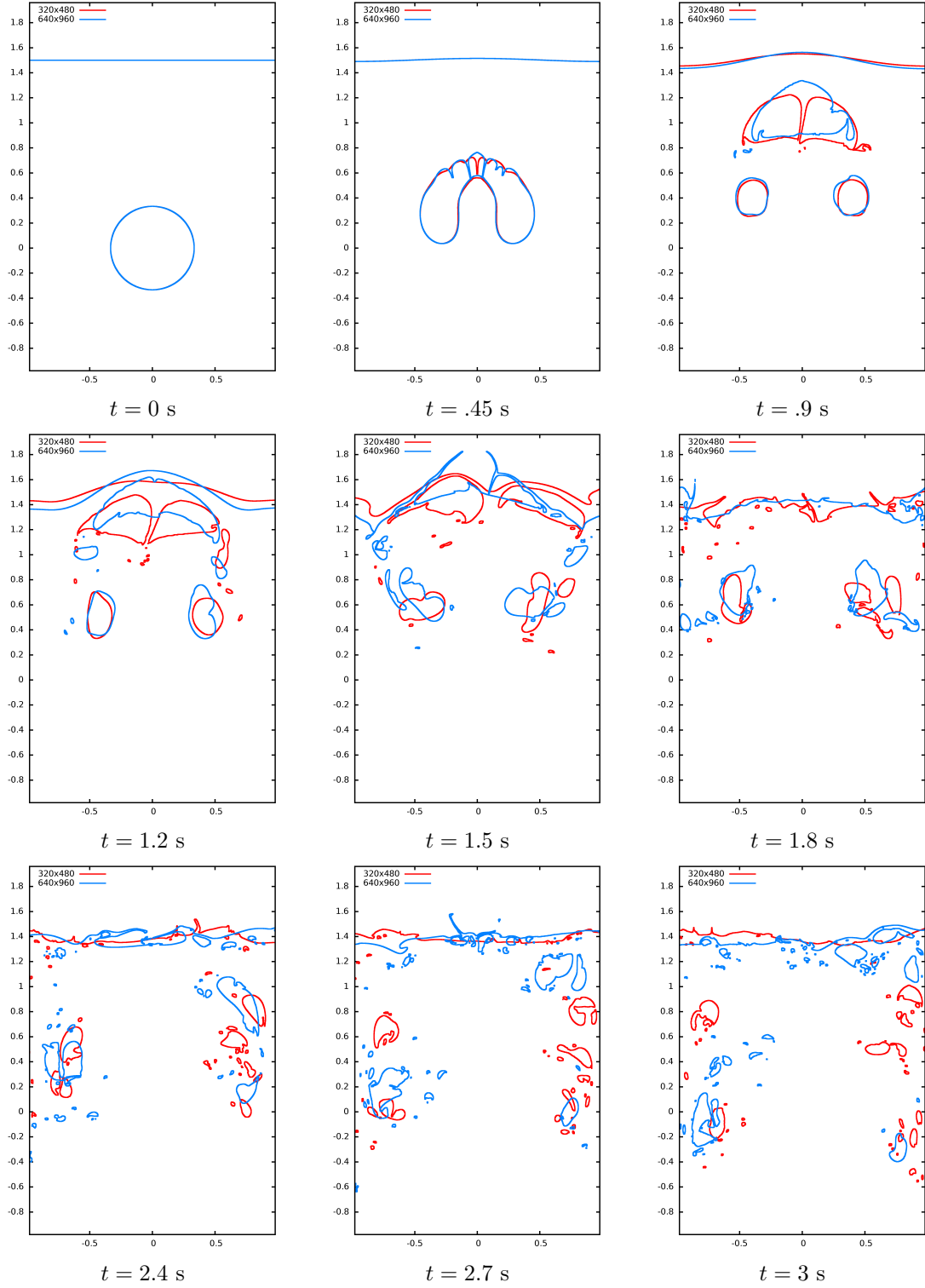


Figure 28: An air bubble rising in a water column with a free surface. The effects of surface tension and viscosity are absent. Note the small scale details that our solver is able to resolve and accurately track over time.

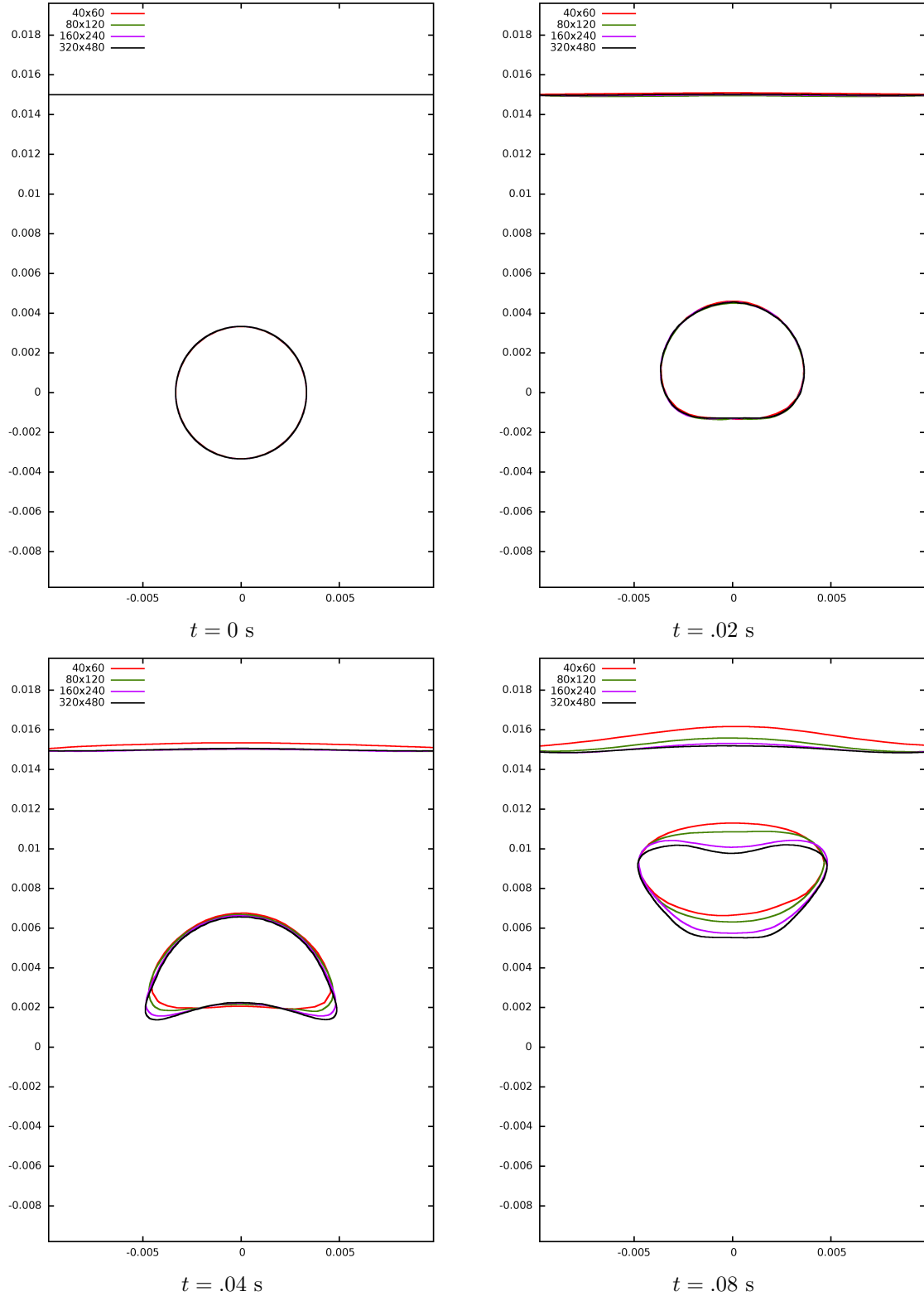


Figure 29: Level set profiles under grid refinement for an air bubble rising in a water column with a free surface in a small computational domain. The effects of surface tension and viscosity are present.

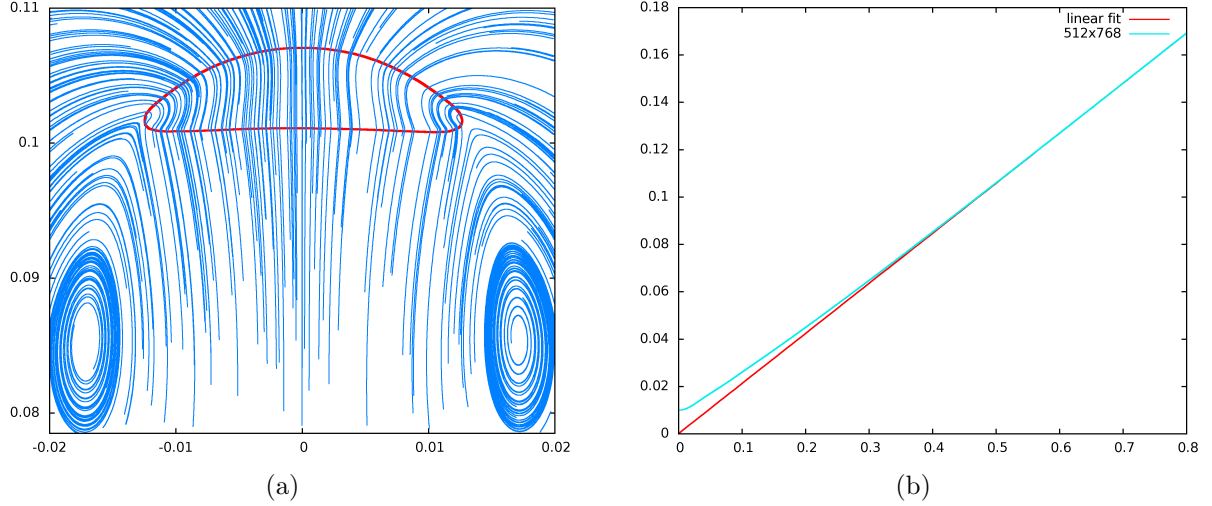


Figure 30: Rising bubble example from Fig. 1A of [16], (a) computed steady state bubble shape and the streamlines inside the bubble and those in its wake, and (b) time evolution of the position of the center of mass of the bubble.

located at $(-0.6 + .4k, .5)$, where $k = \{0, 1, 2, 3\}$ and the other five objects located at $(-0.8 + .4k, .9)$, where $k = \{0, 1, 2, 3, 4\}$. Figure 31 shows the level set at $t = 0, t = .25, t = .5, t = .75, t = 1, t = 1.25, t = 1.5, t = 1.75$ and $t = 2.1$ seconds for two grids of resolutions 320×480 and 640×960 respectively. As can be seen, the objects break up the bubble into a large number of small bubbles which our solver is able to resolve and efficiently track over time. We also show an example where the larger bubbles deform while the smaller bubbles are able to preserve their shape due to surface tension forces by reducing the computational domain to $[-.1 \text{ m}, .1 \text{ m}] \times [-.1 \text{ m}, .2 \text{ m}]$, the radius of the bubble to $1/30 \text{ m}$ with initial density 1.24 kg/m^3 , and adding the effects of surface tension and viscosity with coefficients $\sigma = .0728 \text{ kg/s}^2$ and $\mu = .001137 \text{ kg/ms}$. Figure 32 shows the level set at $t = 0, t = .1, t = .2, t = .3, t = .4, t = .5, t = .6, t = .7$ and $t = .8$ seconds for two grids of resolutions 160×240 and 320×480 respectively. Note that the smaller bubbles remain spherical because of larger surface tension forces while the larger bubbles readily deform. In order to show convergence under grid refinement in the presence of objects, we reduced the computational domain to $[-.02 \text{ m}, .02 \text{ m}] \times [-.02 \text{ m}, .04 \text{ m}]$ and the radius of the bubble to $1/150 \text{ m}$, and reduced the number of objects to five. Two objects are located at $(-.007 + .014k, .01)$, where $k = \{0, 1\}$ and the other three are located at $(-.014 + .014k, .018)$, where $k = \{0, 1, 2\}$. The effects of surface tension and viscosity are present with the same coefficient values as above. The initial density of the air bubble is 1.229 kg/m^3 . Figure 33 shows the level set at $t = 0, t = .075, t = .16$ and $t = .225$ seconds.

6.1.2. Three spatial dimensions

Consider a computational domain of $[-1 \text{ m}, 1 \text{ m}] \times [-1 \text{ m}, 2 \text{ m}] \times [-1 \text{ m}, 1 \text{ m}]$ which is initially filled with water with density 1000 kg/m^3 where the free surface is located at $y = 1.5 \text{ m}$. An air bubble of radius $1/3 \text{ m}$ and density 1.364 kg/m^3 is centered at the origin. The effects of surface tension and viscosity are absent. To break up the bubble into a large number of smaller bubbles, the domain also has 25 spherical objects of radius $.15 \text{ m}$ centered at $(-.8 + .4i, .5, -.8 + .4j)$, where $i, j = \{0, 1, 2, 3, 4\}$. Figure 34 shows the level set at $t = 0, t = .7, t = 1$ and $t = 1.4$ seconds simulated using a grid of resolution $256 \times 384 \times 256$. A close up of the level set at $t = 1$ seconds is shown in Figure 35 illustrating the large amount of topological detail that our solver is able to resolve and accurately track over time.

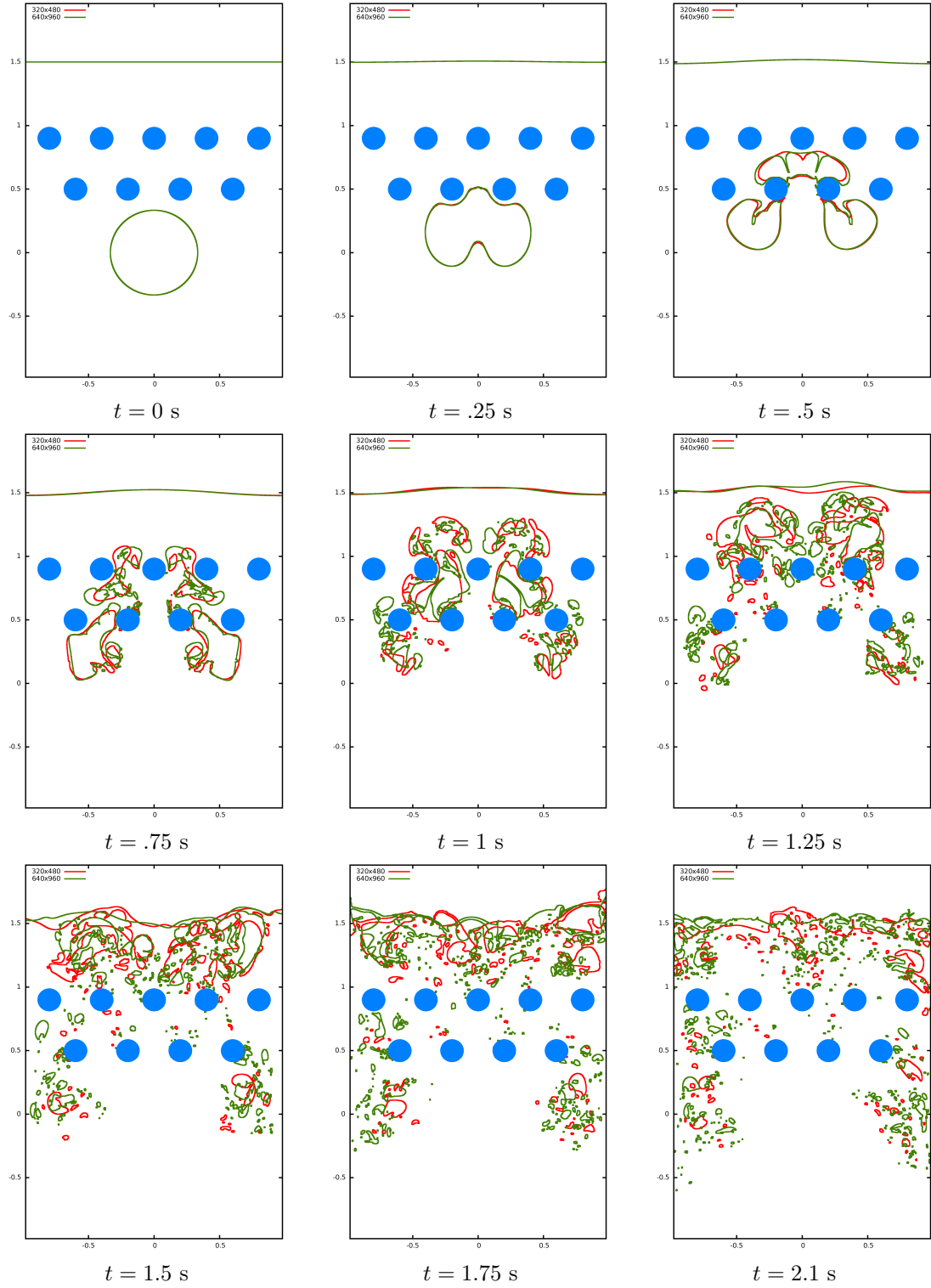


Figure 31: An inviscid air bubble rising in a water column with a free surface in the presence of objects. The objects break up the bubble into a large number of small bubbles which our solver is able to resolve and efficiently track over time.

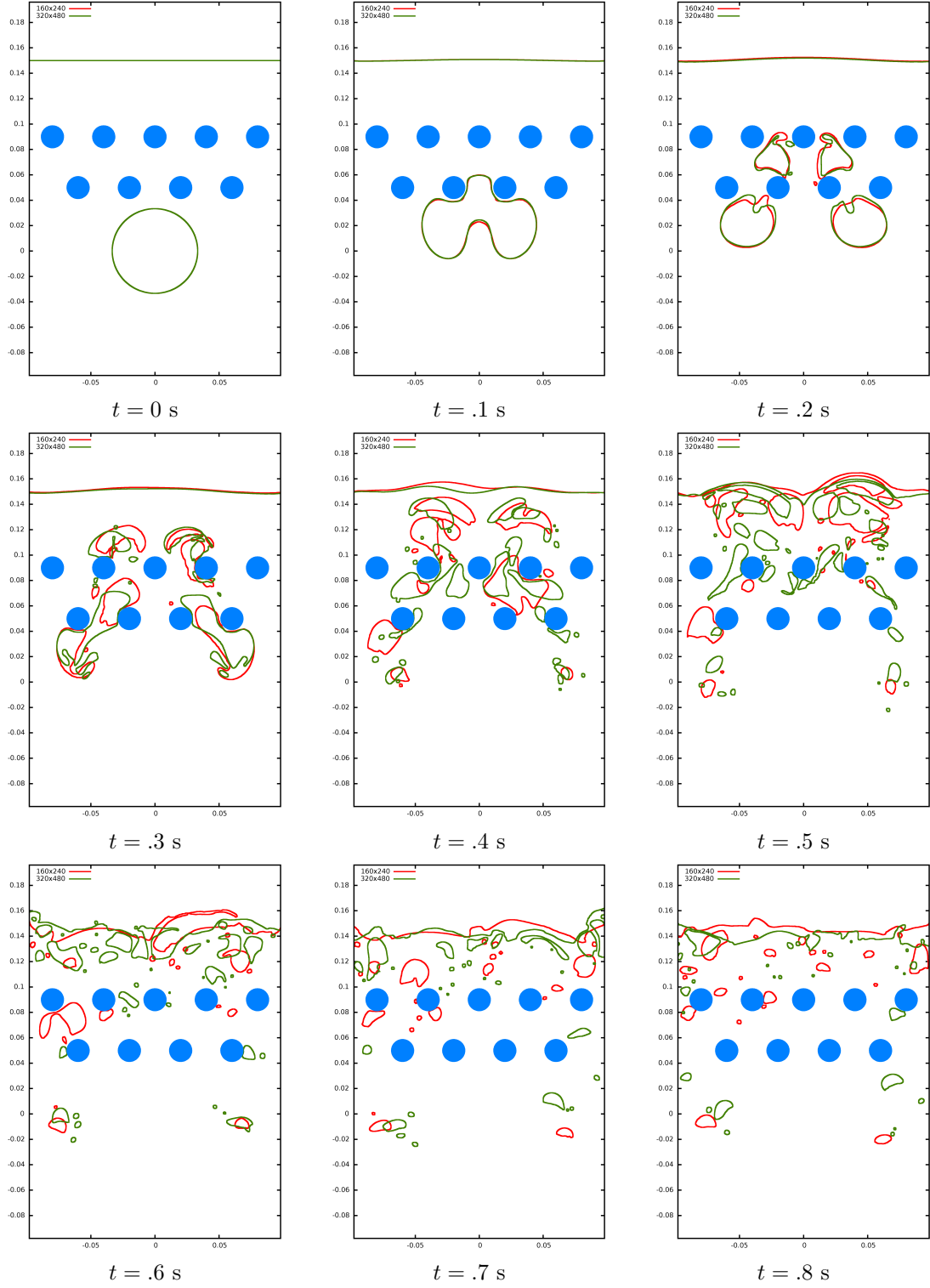


Figure 32: An air bubble rising in a water column with a free surface in the presence of objects in a medium-sized computational domain. The effects of surface tension and viscosity are present. Note that the smaller bubbles remain spherical because of larger surface tension forces while the larger bubbles readily deform.

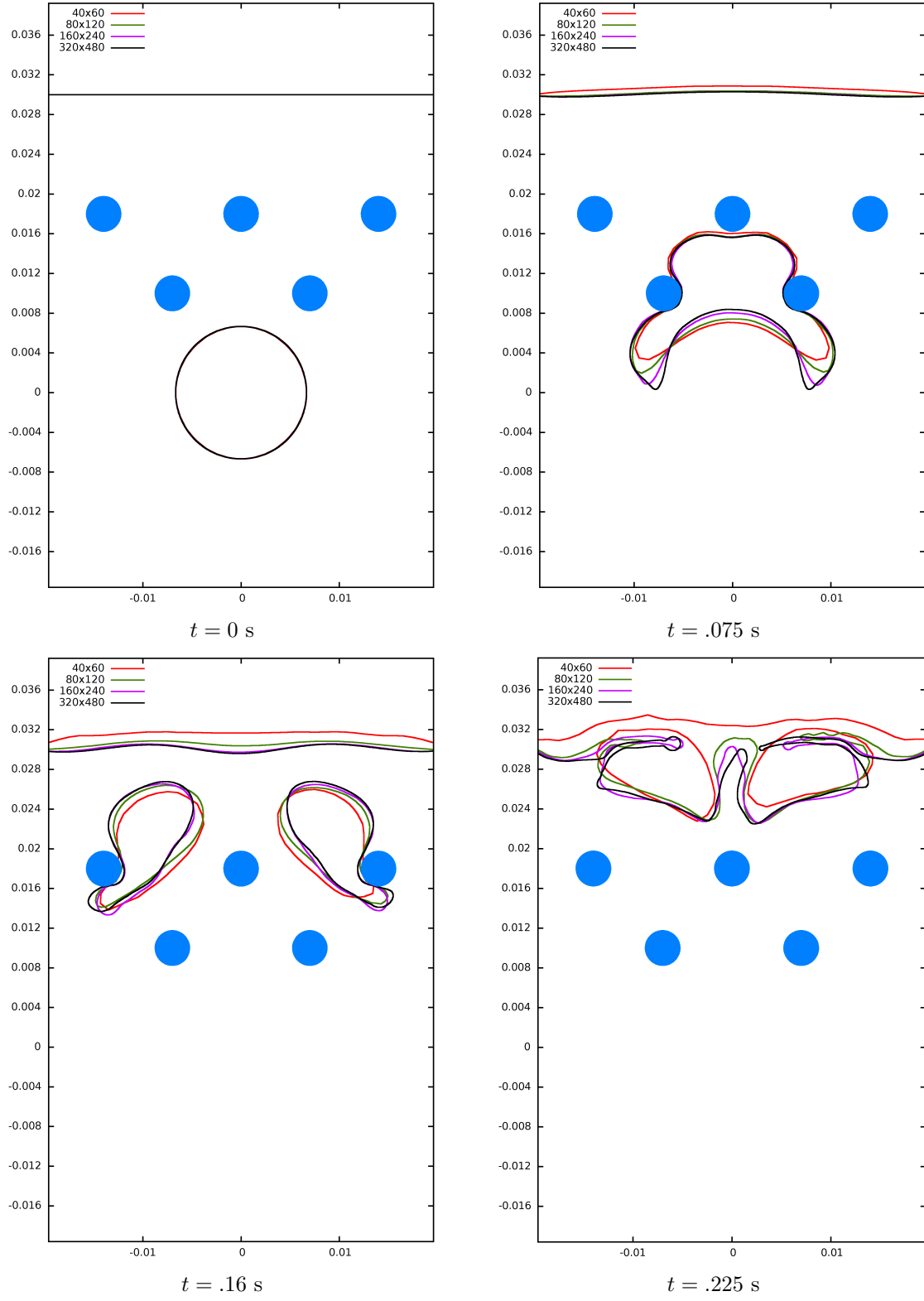
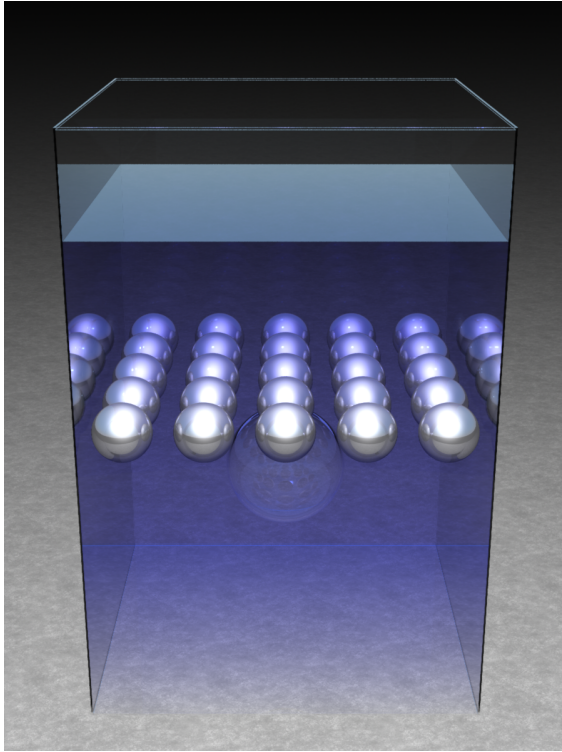
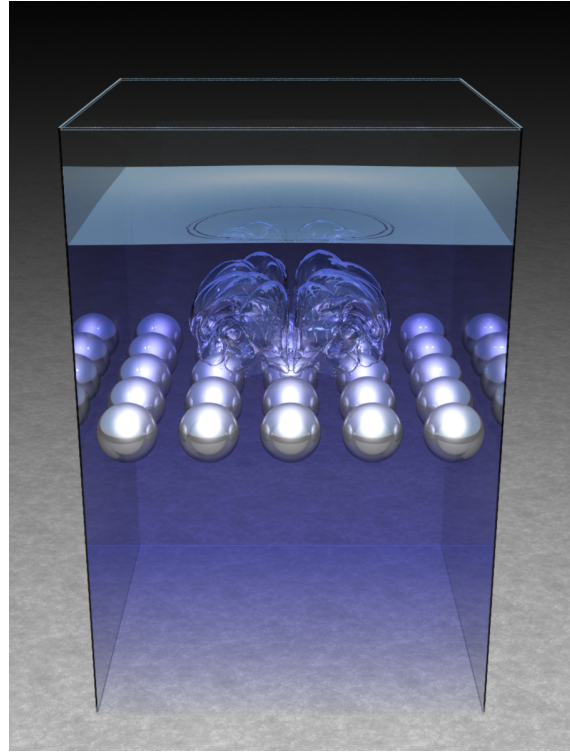


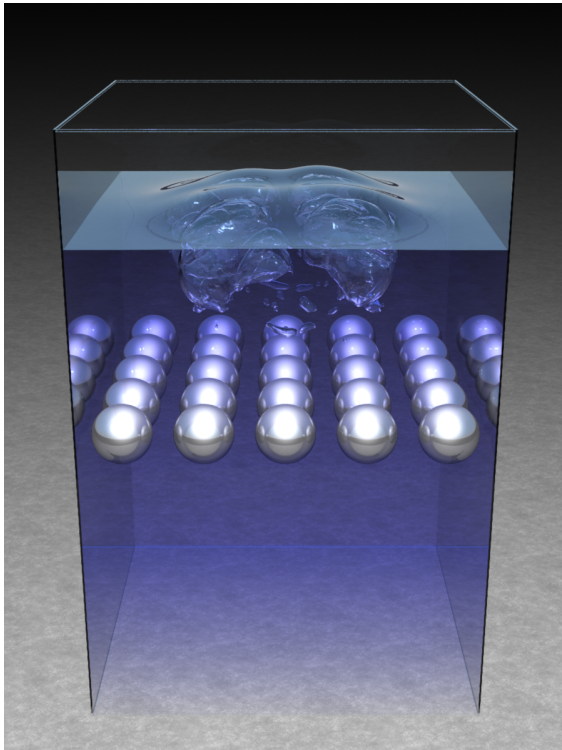
Figure 33: Level set profiles under grid refinement for an air bubble rising in a water column with a free surface in the presence of objects in a small computational domain. The effects of surface tension and viscosity are present.



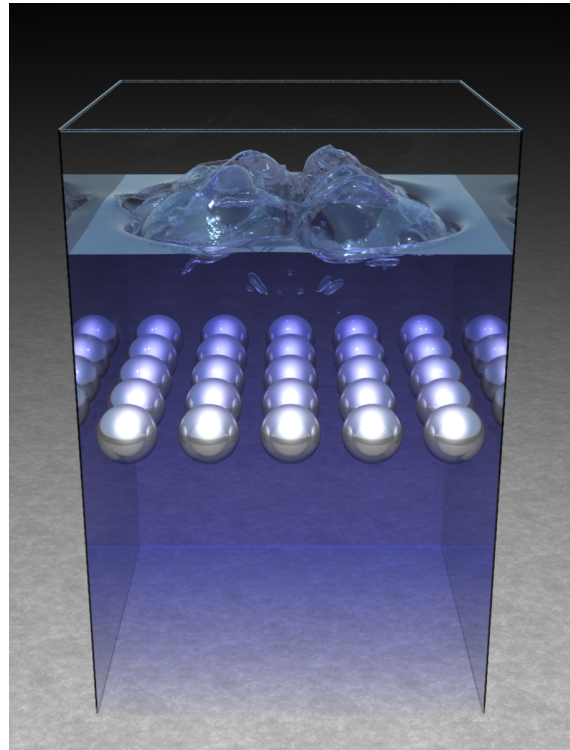
$t = 0 \text{ s}$



$t = .7 \text{ s}$



$t = 1 \text{ s}$



$t = 1.4 \text{ s}$

Figure 34: An air bubble rising in a water column with a free surface in the presence of objects in three spatial dimensions. The effects of surface tension and viscosity are absent.

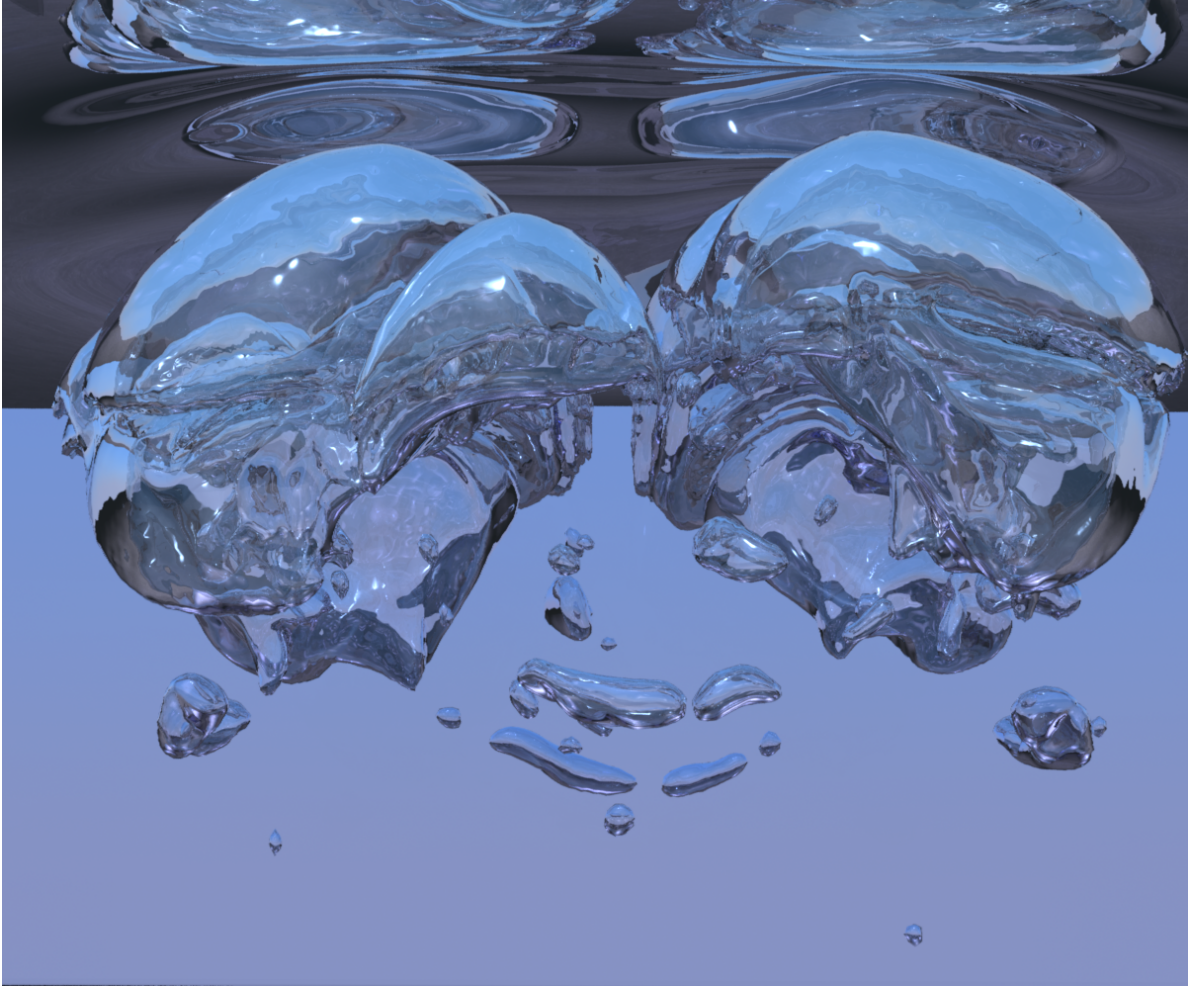


Figure 35: A close up of the three dimensional rising bubbles at $t = 1$ seconds (see Figure 34(c)). Note the large amount of topological detail that our solver is able to resolve and accurately track over time.

7. Conclusion

We designed a method for simulating air bubbles in free surface incompressible flows. To formulate our method, we first proposed a straightforward partitioned solver based on mass tracking. We showed that such an approach suffers from stability issues which have characteristics similar to partitioned (as opposed to monolithic) methods for solid-fluid coupling [13, 26, 12]. These issues can be alleviated using outer iterations on the partitioned solver, although the computation time increases drastically because each time step can require ten or more Poisson solves. Hence, we took a monolithic approach for the air-water problem similar to the solid-fluid coupling in [13, 26, 12] as motivated by [19]. To design this approach, we revisited the partitioned solver of [5] for coupling compressible and incompressible flow and devised a monolithic solver using the ideas from [19] to couple together incompressible flow with fully non-linear compressible flow including shocks and rarefactions. We then simplified this approach greatly to make this approach in line with our straightforward partitioned approach for simulating bubbles. This was achieved by setting both the bubble density and the bubble pressure to be spatially constant, although time-varying. We demonstrated the accuracy and robustness of this method on test problems as well as more realistic problems in both one, two and three spatial dimensions. In the future, we would like to couple this method with those for modeling

cavitation [4] to simulate real-world problems such as ship propellers or breaking waves.

Acknowledgements

Research supported in part by ONR N00014-09-1-0101, ONR N00014-11-1-0027, ONR N00014-11-1-0707, ARL AHPCRC W911NF-07-0027, and the Intel Science and Technology Center for Visual Computing. Computing resources were provided in part by ONR N00014-05-1-0479. M. A. was supported in part by the Nokia Research Center.

Bibliography

- [1] D. Adalsteinsson and J. Sethian. The fast construction of extension velocities in level set methods. *J. Comput. Phys.*, 148:2–22, 1999.
- [2] K. Atkinson. *An Introduction to Numerical Analysis*. John Wiley and Sons, 1989.
- [3] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modelling surface tension. *J. Comput. Phys.*, 100:335–353, 1992.
- [4] Christopher E. Brennen. *Cavitation and Bubble Dynamics*. Oxford University Press, USA, 1995.
- [5] R. Caiden, R. Fedkiw, and C. Anderson. A numerical method for two phase flow consisting of separate compressible and incompressible regions. *J. Comput. Phys.*, 166:1–27, 2001.
- [6] A.J. Chorin. Numerical solution of the Navier-Stokes Equations. *Math. Comp.*, 22:745–762, 1968.
- [7] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *J. Comput. Phys.*, 183:83–116, 2002.
- [8] D. Enright, F. Losasso, and R. Fedkiw. A fast and accurate semi-Lagrangian particle level set method. *Computers and Structures*, 83:479–490, 2005.
- [9] R. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J. Comput. Phys.*, 152:457–492, 1999.
- [10] R. Fedkiw, X.-D. Liu, and S. Osher. A general technique for eliminating spurious oscillations in conservative schemes for multiphase and multispecies euler equations. *Int. J. Nonlinear Sci. and Numer. Sim.*, 3:99–106, 2002.
- [11] F. Gibou, R. Fedkiw, L.-T. Cheng, and M. Kang. A second-order-accurate symmetric discretization of the Poisson equation on irregular domains. *J. Comput. Phys.*, 176:205–227, 2002.
- [12] J. Grétarsson and R. Fedkiw. Fully conservative, robust treatment of thin shell fluid-structure interactions in compressible flows. (*submitted*), 2012.
- [13] J.T. Grétarsson, N. Kwatra, and R. Fedkiw. Numerically stable fluid-structure interactions between compressible flow and solid structures. *J. Comput. Phys.*, 230:3062–3084, 2011.
- [14] E. Guendelman, A. Selle, F. Losasso, and R. Fedkiw. Coupling water and smoke to thin deformable and rigid shells. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 24(3):973–981, 2005.
- [15] Denis Gueyffier, Jie Li, Ali Nadim, Ruben Scardovelli, and Stphane Zaleski. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational Physics*, 152(2):423 – 456, 1999.
- [16] J. G. Hnat and J. D. Buckmaster. Spherical cap bubbles and skirt formation. *Physics of Fluids*, 19:182–194, 1976.

- [17] J.-M. Hong, T. Shinar, M. Kang, and R. Fedkiw. On boundary condition capturing for multiphase interfaces. *J. Sci. Comput.*, 31:99–125, 2007.
- [18] M. Kang, R. Fedkiw, and X.-D. Liu. A boundary condition capturing method for multiphase incompressible flow. *J. Sci. Comput.*, 15:323–360, 2000.
- [19] N. Kwatra, J. Su, J.T. Grétarsson, and R. Fedkiw. A method for avoiding the acoustic time step restriction in compressible flow. *J. Comput. Phys.*, 228(11):4146–4161, 2009.
- [20] M. Lentine, M. Cong, S. Patkar, and R. Fedkiw. Simulating free surface flow with very large time steps. In *ACM SIGGRAPH/Eurographics Symp. on Comput. Anim. 2012*, pages 107–116, 2012.
- [21] M. Lentine, J.T. Grétarsson, and R. Fedkiw. An unconditionally stable fully conservative semi-lagrangian method. *J. Comput. Phys.*, 230:2857–2879, 2011.
- [22] X.-D. Liu, R. Fedkiw, and M. Kang. A boundary condition capturing method for Poisson’s equation on irregular domains. *J. Comput. Phys.*, 154:151, 2000.
- [23] F. Losasso, R. Fedkiw, and S. Osher. Spatially adaptive techniques for level set methods and incompressible flow. *Computers and Fluids*, 35:995–1010, 2006.
- [24] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2002. New York, NY.
- [25] N. Rasmussen, D. Enright, D. Nguyen, S. Marino, N. Sumner, W. Geiger, S. Hoon, and R. Fedkiw. Directable photorealistic liquids. In *Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, pages 193–202, 2004.
- [26] A. Robinson-Mosher, C. Schroeder, and R. Fedkiw. A symmetric positive definite formulation for monolithic fluid structure interaction. *J. Comput. Phys.*, 230:1547–66, 2011.
- [27] A. Selle, R. Fedkiw, B. Kim, Y. Liu, and J. Rossignac. An Unconditionally Stable MacCormack Method. *J. of Sci. Comp.*, 35(2):350–371, 2008.
- [28] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes. *J. Comput. Phys.*, 77:439–471, 1988.
- [29] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes II (two). *J. Comput. Phys.*, 83:32–78, 1989.
- [30] M. Sussman. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *J. Comput. Phys.*, 187:110–136, 2003.
- [31] M. Sussman and P. Smereka. Axisymmetric free boundary problems. *J. Fluid Mech.*, 341:269–294, 1997.
- [32] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, 114:146–159, 1994.
- [33] S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multifluid flows. *J. Comput. Phys.*, 100:25–37, 1992.

Appendix

Consider an infinitesimal element with volume $d\Omega$ and force per unit volume ∇p , implying that the total force on this element is $\nabla p d\Omega$. The work done in displacing this element in an infinitesimal time interval dt is given by $\nabla p d\Omega \cdot d\vec{x} = \nabla p d\Omega \cdot \vec{v} dt$. The total work done dW in the time interval dt is the integral of this work over the entire domain, i.e.,

$$dW = \int_{\Omega} \nabla p \cdot \vec{v} d\Omega dt = \int_{\Omega} \nabla \cdot (p\vec{v}) d\Omega dt \quad (60)$$

since $\nabla \cdot \vec{v} = 0$. Using the divergence theorem, this integral is equivalent to the following surface integral

$$dW = \oint_{\partial\Omega} p\vec{v} \cdot \vec{n} dS dt \quad (61)$$

which we use in the following subsections.

Appendix-I

Consider the one dimensional oscillating bubble problem introduced in Section 2.2 and Figure 1. Since the system is symmetric about the midpoint we only consider the right-half of the domain. In one spatial dimension, water being incompressible has a spatially constant velocity. Since there is no mass transfer between the air bubble and the water this is also the velocity of the bubble-water interface, or the rate of change of the radius of the bubble. Thus at time t , $v(t) = \dot{R}(t)$ at the interface of the bubble of radius $R(t)$. If l is the length of the water region, then $\rho_I l$ is its mass. Let $p_b(t)$ be the pressure inside the bubble and $p_{\text{atm}}(t)$ be the pressure in the air at time t . The total force on the water is given by $p_b(t) - p_{\text{atm}}(t)$. This must equal the mass times the acceleration $\ddot{R}(t)$ of water, i.e.,

$$p_b(t) - p_{\text{atm}}(t) = \rho_I l \ddot{R}(t) \quad (62)$$

Using an equation of state $p = B\rho$, it follows that $p_b(t) = B\rho_b(t) = BM/R(t)$, where M is the constant mass of the bubble. Substituting this in equation (62) gives

$$\ddot{R}(t) = \frac{1}{\rho_I l} \left(\frac{BM}{R(t)} - p_{\text{atm}}(t) \right) \quad (63)$$

Appendix-II

Consider the oscillating bubble problem in two spatial dimensions as shown in Figure 2. Since the total volume of water is conserved, the radius of the water sphere $R_w(t)$ is dependent on the radius of the bubble $R(t)$. Let R^0 be the initial radius of the bubble and R_w^0 be that of the water drop. Then conservation of volume yields

$$\begin{aligned} \pi((R_w^0)^2 - (R^0)^2) &= \pi(R_w(t)^2 - R(t)^2) \\ R_w(t) &= \sqrt{(R_w^0)^2 + R(t)^2 - (R^0)^2} \\ R_w(t) &= \sqrt{a^2 + R(t)^2}, \text{ where } a^2 = (R_w^0)^2 - (R^0)^2 \end{aligned} \quad (64)$$

In fact, the volume is conserved for any annulus with inner radius $R(t)$ and outer radius r , so that $\pi(r^2 - R(t)^2)$ is constant and

$$r\dot{r} = R(t)\dot{R}(t) \quad (65)$$

At any point in time the kinetic energy of an annulus of infinitesimal thickness dr and radius r is given by $\frac{1}{2}(\rho_I 2\pi r dr) \cdot v(r)^2$, where $v(r) = \dot{r} = R(t)\dot{R}(t)/r$ from equation (65). The total kinetic energy of water is obtained by integrating this from the radius of the bubble $R(t)$ to the outer radius of water $R_w(t)$, i.e.,

$$K.E.(t) = \rho_I \pi \int_{R(t)}^{R_w(t)} (R(t)\dot{R}(t)/r)^2 r dr = \pi \rho_I (R(t)\dot{R}(t))^2 \ln(R_w(t)/R(t)) \quad (66)$$

Equating the work done with the kinetic energy, and noting that equation (61) applied to our annulus gives two terms of the form $2\pi r\dot{r}$ results in

$$\int_0^t p_b(\tau) 2\pi R(\tau) \dot{R}(\tau) d\tau - p_{\text{atm}}(\tau) 2\pi R_w(\tau) \dot{R}_w(\tau) d\tau = \pi \rho_I (R(t) \dot{R}(t))^2 \ln(R_w(t)/R(t)) \quad (67)$$

From equation (65), $R_w(t) \dot{R}_w(t) = R(t) \dot{R}(t)$ for any time t and thus

$$2 \int_0^t (p_b(\tau) - p_{\text{atm}}(\tau)) R(\tau) \dot{R}(\tau) d\tau = \rho_I (R(t) \dot{R}(t))^2 \ln(\sqrt{a^2 + R(t)^2}/R(t)) \quad (68)$$

Differentiating both sides with respect to t and simplifying gives

$$p_b(t) - p_{\text{atm}}(t) = \rho_I (\ddot{R}(t) R(t) + \dot{R}(t)^2) \ln(\sqrt{a^2 + R(t)^2}/R(t)) - .5a^2 \rho_I \dot{R}(t)^2 / (a^2 + R(t)^2) \quad (69)$$

Substituting $p_b(t) = B\rho_b(t) = BM/(\pi R(t)^2)$ and rearranging gives

$$\rho_I \ddot{R}(t) R(t) \ln\left(\frac{\sqrt{a^2 + R(t)^2}}{R(t)}\right) + \rho_I \dot{R}(t)^2 \left(\ln\left(\frac{\sqrt{a^2 + R(t)^2}}{R(t)}\right) - \frac{.5a^2}{a^2 + R(t)^2} \right) - \frac{BM}{\pi R(t)^2} + p_{\text{atm}}(t) = 0 \quad (70)$$

Appendix-III

Finally, consider the three dimensional case. Equation (64) becomes

$$R_w(t) = \sqrt[3]{a^3 + R(t)^3}, \text{ where } a^3 = (R_w^0)^3 - (R^0)^3 \quad (71)$$

equation (65) becomes

$$r^2 \dot{r} = R(t)^2 \dot{R}(t) \quad (72)$$

and equation (66) becomes

$$K.E.(t) = 2\pi \rho_I \int_{R(t)}^{R_w(t)} (R(t)^2 \dot{R}(t)/r^2)^2 r^2 dr = 2\pi \rho_I R(t)^4 \dot{R}(t)^2 (1/R(t) - 1/R_w(t)) \quad (73)$$

Equating the work done with the kinetic energy, and noting that equation (61) applied to our thickened shell gives two terms of the form $4\pi r^2 \dot{r}$ results in

$$\int_0^t p_b(\tau) 4\pi R(\tau)^2 \dot{R}(\tau) d\tau - p_{\text{atm}}(\tau) 4\pi R_w(\tau)^2 \dot{R}_w(\tau) d\tau = 2\pi \rho_I R(t)^4 \dot{R}(t)^2 (1/R(t) - 1/R_w(t)) \quad (74)$$

From equation (72), $R_w^2(t) \dot{R}_w(t) = R^2(t) \dot{R}(t)$ for any time t and thus

$$2 \int_0^t (p_b(\tau) - p_{\text{atm}}(\tau)) R(\tau)^2 \dot{R}(\tau) d\tau = \rho_I R(t)^4 \dot{R}(t)^2 \left(1/R(t) - 1/\sqrt[3]{a^3 + R(t)^3} \right) \quad (75)$$

Differentiating both sides with respect to t and simplifying gives

$$p_b(t) - p_{\text{atm}}(t) = \rho_I \ddot{R}(t) \left(R(t) - \frac{R(t)^2}{\sqrt[3]{a^3 + R(t)^3}} \right) + \rho_I \dot{R}(t)^2 \left(\frac{3}{2} + \frac{R(t)^2}{2\sqrt[3]{a^3 + R(t)^3}} - \frac{2R(t)}{\sqrt[3]{a^3 + R(t)^3}} \right) \quad (76)$$

Substituting $p_b(t) = B\rho_b(t) = BM/(\frac{4}{3}\pi R(t)^3)$ and rearranging gives

$$\rho_I \ddot{R}(t) \left(R(t) - \frac{R(t)^2}{2\sqrt[3]{a^3 + R(t)^3}} \right) + \rho_I \dot{R}(t)^2 \left(\frac{3}{2} + \frac{R(t)^2}{2\sqrt[3]{a^3 + R(t)^3}} - \frac{2R(t)}{\sqrt[3]{a^3 + R(t)^3}} \right) - \frac{BM}{\frac{4}{3}\pi R(t)^3} + p_{\text{atm}}(t) = 0 \quad (77)$$